MODIFICATION ON THE SCHOFIELD STOCHASTIC MODEL FOR ESTUARIES

by

Nicholas John Manthos

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Statistics

APPROVED:

Richard G. Krutchkoff, Chairman

Dinshaw N. Contractor

Walter R. Pirie

February, 1976

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author would like to thank Dr. Richard G. Krutchkoff for the opportunity to write this thesis and his continual direction and patience throughout this work.

Special thanks to Dr. Raymond H. Myers for financial assistance supplied by the Statistical Consulting Laboratory.

The author is obliged to Dr. Walter R. Pirie and Dr. Dinshaw N. Contractor for taking time from their schedules to serve as committee members and to Dr. Keith Lee for having supplied the routine for the incomplete gamma.

I would sincerely like to extend appreciation to Mrs. Mollissa Faye Roop for her excellent and efficient typing of this thesis.

Finally special indebtedness to M.S.F. who truly made life enjoyable during this time.

TABLE OF CONTENTS

Page

Chapter

PREFACE

One method of predicting the effects of sewage and industrial waste which is being used is the mathematical model. Presently with the increased understanding of the interactions between chemical and biological components, more complex and detailed mathematical models have been developed to model the effect of waste discharge in estuaries. Further work is still needed to provide better models and to refine old ones.

The model by Schofield and Krutchkoff (5) is one of the most efficient in existence but is not currently being used effectively due to its complicated structure. The model was designed to handle twelve components and cannot easily be used for fewer components.

The model was originally developed as a general deterministic model for an estuary, which allows the estuary to be segmented into any number of parts with conditions varying from segment to segment. The model considers twelve biological and chemical components and provides one dimensional predictions of concentrations of the twelve components, in time and position. It does this by solving twelve simultaneous differential equations of the form

$$\frac{\partial C_i(x,t)}{\partial t} = \frac{1}{A(x,t)} \frac{\partial}{\partial x} [E(x,t)A(x,t) \frac{\partial C_i(x,t)}{\partial x} - V(x,t)A(x,t)C_i(x,t)]$$

$$- \frac{C_i(x,t)}{A(x,t)} \frac{\partial A(x,t)}{\partial t} + R_i(x,t,\underset{\sim}{c}) \qquad (5)$$

where

$A(x,t)$	is the cross-sectional area of the estuary as a function of position and time
$C_i(x,t)$	is the concentration of component i as a function of position and time
$E(x,t)$	is the diffusion coefficient as a function of position and time
$V(x,t)$	is the average cross-sectional fluid velocity as a function of position and time
$R_i(x,t,\underset{\sim}{c})$	is the set rate of increase of $C_i(x,t)$ due to all sources and sinks as a function of position, time and of all the other component concentrations

and the i components are:

1. Organic carbon
2. Inorganic carbon
3. Organic nitrogen
4. Ammonia
5. Nitrite and nitrate
6. Phosphorous
7. Oxygen deficit
8. Phytoplankton (algae)
9. Protozoa
10. Zooplankton
11. Higher predators
12. Bacteria (nitrosomonas)

No analytical solution can be found for these twelve simultaneous differential equations and therefore a computer solution was developed. At each grid point an initial concentration is specified for each of the twelve components. This determines the value of everything on the right hand side of the equation thus enabling $\partial C_i/\partial t$ to be found. Since the solution is done by computer a careful selection of the grid spacing must be chosen to assure a convergent solution and at the same time not use excessive computer time.

The Schofield model was originally run for verification purposes on the Potomac estuary and thus certain internal equations and specific rate constants were chosen to pertain to the Potomac. Whenever applying it to another estuary certain features of that particular estuary make it necessary for some internal adjustments in the equations and rate constants.

Harry Bard (1) took the Schofield model and applied it to the James River estuary. He did a sensitivity analysis on some of the parameters in the model. Bard incorporated a common block which contained all pertinent parameters of the main program and its eleven subroutines. This makes all variables the same from one subroutine to the next. In order to model the James, Bard chose to change expressions for the maximum tidal velocity and maximum tidal lage, which are functions of the grid point position and thus depend on the particular estuary modeled. He also chose to replace the mathematical expression for the average cross-sectional area with data for the cross-sectional area at each grid point. The calculations of fluid velocity and volumetric flow rates as a function of freshwater flow rate, position and tidal phase were also changed to fit the James. His changes made the program more applicable for the James River. Similar changes must be undertaken in modeling other estuaries, in order to adapt the program to fit the estuary intended.

Sandra DePietro (2) took the Bard version of the program and made further changes to make it even more general and easier to adapt to a specific estuary. DePietro took expressions for the maximum tidal velocity and tidal phase lag which were in the form of regression equations specific to the estuary modeled and used these. The maximum tidal velocity was as follows:

$$MTV = A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4 \tag{2}$$

where

MTV = maximum tidal velocity (mi/hr)

A_0, A_1, A_2, A_3, A_4 = appropriate constants

X = position (mi).

The tidal phase lag was

$$L = G_0 + G_1X + G_2X^2 \tag{2}$$

where

L = phase (hr)

G_0, G_1, G_2 = appropriate constants

X = position (mi) .

The values of the constants vary with each estuary and can be estimated from information given in tide table surveys.

She also changed expressions for oxygen reaeration rate which enabled the user to choose the one which best fits the modeled estuary because of differences in estuary depth and velocity. The following equations were used:

$$K_2 = \frac{12.9\mu^{1/2}}{H^{3/2}}$$

estuary depths of 1-30 feet, velocities of .5-1.6 ft/sec.

$$K_2 = \frac{11.6}{H^{1.67}}$$

estuary depths of 2-11 feet, velocities of 1.8-5.0 ft/sec.

$$K_2 = \frac{21.6\mu^{0.67}}{H^{1.85}}$$

estuary depths 4-11 feet, velocities of .1-5.0 ft/sec.

$$K_2 = \frac{(D\cdot\mu)^{.5}}{H^{1.5}}$$

for estuaries not in above three categories

where

μ - tidal velocity (ft/sec)

H - depth (ft)

D - diffusivity of oxygen in water (.000081 ft^2/hr) .

In addition reaeration rates were also made functions of wind velocity. The reaeration rate increased 25% with every 10 knot wind increase.

The major obstacle to practical utilization of the model is that it is necessary to have all twelve components in the model in order to use it. Many times, however, only a subset of the twelve are available. The concentrations are found by solving twelve simultaneous differential equations, which are interlinked by the concentration of one component being dependent on the concentrations of some of the others, thus reducing the number of equations requiring adjustments in these dependencies. The equations are linked by the term $R_i(x,t,\underset{\sim}{c})$, the rate of increase due to all sources and sinks of the i^{th} concentration. It is these which must be altered to accommodate the use of less than twelve equations. One accomplishes this by assigning default values for components not wanted in the model and solving the reduced number of equations. As an example, consider taking one of the equations, say ammonia. In the program the concentration of algae is found by first evaluating the terms

$$\frac{1}{A(x,t)}\ \frac{\partial}{\partial x}\ [E(x,t)\ \frac{\partial C_i(x,t)}{\partial x} - V(x,t)A(x,t)C_i(x,t)]$$

and setting them equal to $\partial C/\partial x$. Then $\partial C/\partial t$ is found by computing

$$\partial C/\partial t = \partial C/\partial x + R_i(x,t,\underset{\sim}{c}).$$

In the case of ammonia this equation yields

$$\partial C/\partial t = \partial C/\partial x + FRA(4)*K(4,J) + K12*C(3,IA,J) - 20.0*K(35,J)$$

where

FRA(4) = fraction of ammonia in sewage discharge

K(4,J) = time variable pollution discharge rate (lb/mi^3hr)

K12 = rate coefficient of conversion of organic nitrogen to ammonia (hr^{-1})

C(3,IA,J) = concentration of organic nitrogen

K(35,J) = growth rate for bacteria (lb/mi^3hr) .

In this equation it can be seen that everything is known when all twelve components are in the model. However, if one drops some of the components then some of these terms might not be known. If organic nitrogen is dropped along with bacteria then the amount of organic nitrogen converted to ammonia is not known and the growth rate for bacteria (given by K(35,J)) = GRO12*C(4,IA,J)/(KM12 + C(4,IA,J)*C(12,IA,J) where

K(35,J) = growth rate for bacteria (lbs/mi^{3}hr)

GRO 12 = growth rate coefficient for bacteria

C(4,IA,J) = concentration of ammonia at grid point j

KM12 = Michaelis-Menten or half saturation concentration of substrate for bacteria (lb/mi^3)

C(12,IA,J) = concentration of bacteria at grid point j

is also not known because the concentrations of organic nitrogen and bacteria, which normally would also have been calculated had they been in the model, are now not known. All other components have similar $R_i(x,t,\underset{\sim}{c})$ terms which can not be evaluated if some components are dropped from the model.

By assigning default values for components not in the equation the unknown quantities can be calculated and the model can run as it does normally. If the built-in default values are not wanted, then the user can supply one of two types of default values depending on what he knows or prefers. One method is to give a component, not in the model, an average concentration over the length of the estuary model (in parts per million) that the user thinks is appropriate for that estuary. This enables certain rate constants to be calculated in the operation of the program. The other method is to supply the rate constants directly for those not in the model. These rate constants are growth, predation and respiration rates of the five biological components, the model, the percentage of organic carbon used per hour, and percentage of organic nitrogen converted to ammonia. As in the case of ammonia, if organic nitrogen was not in the model the term K12*C(3,IA,J) must be specified and if bacteria was not in the model K(35,J) must be specified. For other equations similar values must be given depending on which components are dropped from the model.

In the initial development of the model, the stochastic coefficient Δ_i was used and it was shown that the concentrations followed a scaled Poisson distribution (5). This coefficient Δ_i relates the mean and variance of the i^{th} concentration distribution and is a function of fluid turbulence, concentration, pollution type, system stability and other factors which reflect the degree of randomness. A large value of Δ_i would be expected during periods of process instability or for pollution types with more unpredictable behavior. The relationship between the mean, variance and Δ_i is as follows.

$$\mathrm{Var}_i(\mathrm{concentration}) = \Delta_i \cdot E(\mathrm{concentration})$$

from this Δ_i can be estimated by

$$\Delta_i = \frac{\mathrm{var}_i(\mathrm{concentration})}{E_i(\mathrm{concentration})}$$

where the mean and variance can be estimated from data already collected. Once this was done confidence intervals were found using the Poisson.

However, the Poisson is a discrete distribution while the concentrations are continuous so it was advantageous to approximate the Poisson distribution using a gamma distribution by

$$f(x,\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} e^{-x/\beta} x^{\alpha-1} \qquad 0 < x < \infty \quad \alpha,\beta > 0 .$$

For the gamma distribution the mean and variance are

$$\mu = \alpha\beta$$

and

$$\sigma^2 = \alpha\beta^2$$

By specifying Δ for each of the i components and given the mean (from prediction in the program) we can obtain estimates of α and β by using

$$\alpha_i = \mu/\Delta_i$$

and

$$\beta_i = \Delta_i .$$

The gamma distribution is now completely specified and can be used in finding confidence intervals. This was done by using a program which gave values of the incomplete gamma function and then incrementing until the appropriate endpoints were found for the confidence coefficient specified.

I. PROGRAM USE

This version of the Schofield model may be used to predict the concentrations of up to twelve biological and chemical components in a stream or estuary. Any of the twelve may be selected in any combination desired. The input requirements are broken into four categories, hydrological, geometrical, meteorological and water quality data.

Hydrological data

The hydrological data needed are the freshwater flow rate, tidal phase lag, and maximum tidal velocity as a function of distance along the stretch. For tidal phase lag and maximum tidal velocity these functions are estimated using the regression equation

$$MTV = A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4$$

$$L = G_0 + G_1X + G_2X^2$$

The coefficients in these equations must be supplied for each stretch. They can be estimated from information in tide table surveys.

Geometrical data

This data consists of average cross-sectional area and the maximum fractional deviation of cross-sectional area at high or low tide and depth of the estuary for selected grid points along the stretch.

Meteorological data

This model requires the daily sunlight intensity (langleys) and moon phase on the first day modeled.

Water Quality data

The sewage discharge rate, the benthal demand, and land runoff are needed as well as the water temperature. The internal concentrations, of the components selected in the model, at the selected grid points downstretch are required but are not expected to be known and are obtained from an initial run of the program. (The concentrations of these components are then punched on cards to be used for making the prediction run.)

Miscellaneous data

The number of grid points into which the stretch is divided, the upstretch and downstretch boundaries in miles, initial and final time of day, initial and final day of year modeled and number of simultaneous differential equations in the model are all required. A number of output options can also be selected.

Output Options

I. Graphs of the output on any given day can be requested and the type of axis to be used can be specified. These types are:

1. axis vary day to day
2. axis used on first day used for all days
3. same axis to be used for every day modeled

II. Punched output is available for any of the days modeled or on just the last day modeled.

III. Confidence intervals on the concentrations, with the confidence coefficient an option, can be graphed

IV. Regular output of concentrations of components printed at points downstretch(this is always given).

When first using the program it is necessary to make an initial run to obtain estimates of the pollutant concentrations in the model. This is done by selecting the initial run option in the program and supplying all the necessary data except the component concentrations at the grid points downstretch. Punched output is then obtained to be used in the actual runs for prediction.

II. INPUT DATA

2.1 Data Supplied Through MAIN

The data supplied through MAIN allows one to select the approximate oxygen reaeration rate equation. Two cards are typed beginning in card column seven and are put in following the common statement in program MAIN. For formula

$$K_2 = \frac{12.9\ \mu^{1/2}}{H^{3/2}}$$

H - estuary depths of 1-30 ft

μ - velocities of .5-1.6 ft/sec

use cards K2 = 1

XK2 = 0 .

For formula

$$K_2 = \frac{11.6\ \mu}{H^{1.67}}$$

H - estuary depths of 2-11 ft

μ - velocities of 1.85-5 ft/sec

use cards K2 = 2

XK2 = 0 .

For formula

$$K_2 = \frac{21.6\ \mu^{0.67}}{H^{1.25}}$$

H - estuary depths of 4-11 ft

μ - velocities of .5-5.0 ft/sec

use cards K2 = 3

XK2 = 0 .

For formula

$$K_2 = \frac{(D \cdot \mu)^{.5}}{H^{1.5}}$$

for estuaries not in the above three categories

D = diffusivity of oxygen in water (.000071 ft^2/hr)

use cards K2 = 4

XK2 = 0 .

And for formula

K_2 = user supplied expression or numerical value

use cards K2 = 0

XK2 = user supplied expression or numerical value.

2.2 Data Supplied Through Cards Read in on Units 5, 8, and 9

Data must be supplied by reading in cards at the end of the deck. What components are to be in the model, what type of default values for those components not in the model must be provided as well as the initial run option to reach steady state. One must also specify the type of axis to be used in plotting and whether of not confidence intervals are to be calculated. Also the data regarding the make up of the estuary and sewage input are read in.

The first card is the card which sets up the conditions of the estuary under which the model is to be run. Thirteen variables are to be read in with format (F3,2F10.4,F5.3,I3,F5.3,I3,F5.3,I3,3F10.5,I1)

card column	variable
1-3	NN - number of sections in estuary
4-13	XI - upstream boundary point (mile)
14-23	XO - downstream boundary point (mile)
24-28	TI - initial time of day (hour)
29-31	IDAY - initial day of year
32-36	TM - final time of day (hour)
37-39	MDAY - final day of year
40-44	PRI - number of printouts for each tidal cycle
45-47	IND - number of pollution components model
48-57	TIDE - time of high tide on initial day
58-67	FAZ - moon phase at initial time of day
68-77	TMAX - number of minutes authorized for computer run
78	KP - 1 for punched output on final day 0 for no punched output
79	IWK2 = 0 for no wind = 1 for 10 knot wind = 2 for 20 knot wind = 3 for 30 knot wind = 4 for 40 knot wind = 5 for 50 knot wind = 6 for 60 knot wind where each 10 knot increases, increases reaeration 25%

The next card read is the option card. In card columns one through twelve one selects which components are to be included in the model and whether the default values are internal or user supplied. The components are numbered as follows:

1. organic carbon
2. inorganic carbon
3. organic nitrogen
4. ammonia
5. nitrite and nitrate
6. phosphorus
7. oxygen deficit
8. algae
9. protozoa
10. zooplankton
11. higher predators
12. bacteria.

This ordering must be kept at all times throughout the reading in of data. The first twelve card columns define INDEX(I) as follows:

CC1-12 = 0 if particular component is in model

= 1 if particular component is not in model but default values are internal

= 2 if particular component is not in model but default values are user supplied.

Next in CC13 IDEF is defined as to what type of default values are used

CC13 = 0 if default values are to be used as concentrations in ppm

= 1 if default values are to be used as rate constants.

In CC14 IPLOT is defined and determines which type of axis is to be used in plotting

CC14 = 0 if axis is to be determined from day to day
= 1 if the axis for the first day is to be used for all subsequent days
= 2 if the axis is to be the same for every day and are to be read in.

In CC15 INRUN is defined as to whether an initial run is being made

CC15 = 0 if the program is a regular run for predictions
= 1 if the program is being run to reach steady state with punched output.

If a 1 is chosen then in CC1-12 a 1 is put in those not to be in the model, and IDEF CC13 = 0.

ICON in CC16 is defined to chose the option of confidence intervals

CC16 = 0 no confidence intervals calculated
= 1 confidence intervals calculated.

The options chosen in the first card determines what cards are read in next. If a 2 in any of the CC1-12 appears, a default value must be read in for that component. The number of cards read in here must equal the number of 2's on the option cards.

If IDEF = 0 then default values are read in as ppm for the concentration with a format (F6.4). That is CC1-6 = value of concentration in ppm. These cards must come in the order of the components left out from lowest numbered to highest number component not included.

If IDEF = 1 then default values are read in for rate constants with format (4F6.4) as follows for the five biological components numbered 8-12 left out.

CC1-6 = predation rate

CC7-12 = growth rate

CC13-18 = death rate

CC19-24 = respiration rate.

If CC1 = 2 on the option card and IDEF = 1 then the amount of organic carbon used per hour is read in as format (F12.5) i.e. CC1-12 = amount used. If CC 3 = 2 on the option card and IDEF = 1 then the amount of organic nitrogen converted to ammonia is supplied by format (F12.5) i.e. CC1-12 = amount converted. These cards are read in in the order given above but only if IDEF = 1 and a 2 appears in the card column on the option card in CC8-12, CC1, or CC3. Otherwise they are left out.

Next if NPLOT = 2 the data for the axis are supplied as follows. For each of the components in the model:

XMIN(I) = minimum value that the i^{th} component is expected to reach

SCALF(I) = scale factors of units per inch, on an eight inch graph chosen such that the largest expected value of the i^{th} component will be on the axis with format (2F12.5)

CC1-12 = value of XMIN(I)

CC13-24 = value of SCALF(I)

and the ordering coming once again from lowest numbered component to highest numbered. If NPLOT $\neq$ 2 then these cards are left out.

The next cards are supplied only if ICON = 1. They involve the stochastic coefficient Δ and the confidence coefficient. The value Δ

is an expression which relates the randomness and instability of the component and an estimate can be obtained from previous data by the formula

$$\Delta = \frac{\text{mean of concentration}}{\text{variance of concentration}}$$

The value of Δ and the confidence coefficient are read in on cards with format (2F6.4):

CC1-6 = value of Δ

CC7-12 = value of confidence coefficient.

Once again these cards occur in the order of the components in the model. The next two cards contain the values of the regression coefficients in the equation

$$MTV = A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4$$

$$L = G_0 + G_1X + G_2X^2 \quad .$$

The first card contains the A coefficients with format (5F12.10):

CC1-12 = value of A_0

CC13-24 = value of A_1

CC25-36 = value of A_2

CC37-48 = value of A_3

CC49-60 = value of A_4 .

Next is the G coefficient with format (3F12.10)

CC1-12 = value of G_0

CC13-24 = value of G_1

CC25-36 = value of G_2 .

The next eleven cards give data for determining the volumetric freshwater flow rate (XQ), which tends to increase downstretch as a function of position (Z(I)). The constant increase as well as the number of positions the flow rate will increase is chosen. NCOEF determines how many positions downstretch that will increase up to a maximum of ten. XQCOEF(I) and Z(I) are used as pairs to determine at which grid point (Z(I)) and at what constant increase XQCOEF(I). The number of grid points at which increases will occur NCOEF is specified and then Z(I) through ...Z(NCOEF) will designate at which grid point increases will occur. The grid points must occur in ascending order. The value of XQCOEF(1)...XQCOEF(NCOEF) will specify the multiplicative increase in the flow rate. The remaining pairs Z(NCOEF+1)...Z(10) and XQCOEF(NCOEF+1)...XQCOEF(10) are all set equal to zero. The eleven cards are read in as follows:

<u>card 1</u> format (I2)

CC1-2 = value of NCOEF

<u>cards 2-11</u> format (2F10.6)

CC1-10 = value of Z(I)

CC11-20 = value of XQCOEF(I) .

The next cards identify the twelve components in the model. The titles of these components (see previous listing) are punched in CC1-CC40. They must come in ascending order according to their numbers listed and all twelve supplied.

Sets of inputs each of which is preceded by a "header card" now follow and identify the parameters and the form of the data to be read. The ones considered are cross-sectional area (mi^2), sewage input (lb/day), benthal demand (lb/mi^{3}hr), land runoff (lb/mi^{3}hr), maximum fractional deviation of cross-sectional area, estuary depth (ft) and the concentration of the components selected (ppm). The header card describes the type of data to follow and is in the format (3I5, 5X, IDA4) where

CC1-5	identify the value of IPT which gives the type of data IPT = 1 point source discharge = 2 constant within a segment > 2 continuous
CC6-10	identifies the variable NC which the parameter or pollutant to be considered
CC11-15	identifies ITYPE which distinguishes between a parameter (1) or a pollutant (2)
CC21-60	identifies the complete name of the parameter or pollutant.

If IPT = 1, the cards following will contain a value in units of lb/day and a position in miles with format (2E20.5). Reading will stop when a negative position appears.

If IPT = 2, the cards following will contain a value and a position in miles with format (2E20.5). The grid points which fall between the last position read and the new position are assigned the previously specified value. Reading will stop when a position less than or equal to zero occurs.

If IPT > 2, the cards following will contain continuous data. The numerical value of IPT determines the number of value position pairs to be read. Each card has four pairs of data with format (8F10.6).

For values of IPT = 1 the type of data are sewage input, industrial waste and river point sources. For this category the value of NC read in is determined by the variable K(I,J) (see appendix) for NC = I.

For values of IPT = 2 benthal demand, land runoff and maximum fractional deviation of cross-sectional area data are considered and are found in K(I,J) for the values of NC.

For values of IPT > 2 the twelve components, estuary depth, and cross-sectional area are considered. After all these data sets have been read in a blank card follows to signal the end of reading parameters.

Next come the cards which contain the data for each day modeled. They include daily water temperature, volumetric flow rate, daily sunlight total, options on punched output for that day, Namelist option and plotted output for that day. The format is (3F10.4, 20X, 3I5) where:

Card Column	Variable
1-10	Water temperature (°C)
11-20	Volumetric flow rate (cfs)
21-30	Daily sunlight totals (langleys)
31-50	Skip 20 spaces
51-55	Card punch option
61-65	Plot option .

If the user desires, no temperature may be put in and an equation internal to the program will be used to calculate it.

CC51-55 = any integer other than 0 gives punched output

CC61-65 = any integer other than 0 gives plotted output.

When an initial run is being made to reach steady state, only one card specifying the conditions of the day is read in, and the conditions should be chosen so as to be about an average day for the time of year being modeled. This ends the data to be read in on Unit 5.

Unit 8 is to be used only when comparison of actual data is wanted and must be accompanied by the appropriate job control language for the system being used. A card is read for each position and time data was collected with the following format (I5, F5.0, F5.2, 5X, 6(I2, F8.3)).

Card Column	Variable
1-5	Day of year
6-10	Time (hr)
11-15	Position (mi)
16-20	Blank
21-22, 31-32, 41-42 51-52, 61-62, 71-72	Component number
23-30, 33-40, 43-50 53-60, 63-70, 73-80	Observed value

These cards are followed by a blank indicating the end of the data to be read and are left out if no comparison is to be made.

Next Unit 9 is used to exercise the Namelist option if selected on the data cards used for describing each day modeled, and the appropriate job control language is once again necessary. All variables included in the Namelist may be changed either initially or for a particular day if the option is used. The following cards are used if Namelist option is chosen. The first CC is blank, CC2 is an ampersand (&) followed by the Namelist name (LIST1) in CC3-7. The next space must be blank. Next the variable name = constant for each value to be read appears and is separated by a comma if there is more than one. If values are read into an array, the array name should be written without subscripts, followed by an equal sign and values for all of the array elements and separated by commas. The end of reading is then signaled by &END.

BIBLIOGRAPHY

1. Bard, H. and R.G. Krutchkoff, Predicting Pollution in the James River Estuary, Water Resources Research Center, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, Bulletin 70, 1972.

2. DePietro, S.A., "GEM: Generalized Estuary Model A Variation on the Schofield-Krutchkoff Stochastic Model for Estuaries," Unpublished MS Thesis, Virginia Polytechnic Institute and State University, August, 1975.

3. Dyer, K.R., Estuaries: A Physical Introduction, John Wiley and Sons, Inc., New York, 1973.

4. Nemerow, N.L., Scientific Stream Pollution Analysis, McGraw-Hill Book Co., New York, 1974.

5. Schofield, W.R. and R.G. Krutchkoff, Stochastic Model for a Dynamic Ecosystem, Water Resources Research Center, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, Bulletin 60, 1971

APPENDIX I

PROGRAM DESCRIPTIONS

I.1 MAIN Program

The MAIN program is used to specify the conditions under which the program is to run for the estuary modeled. The user will specify many options he wishes to use and selection of which components he wants in the model. MAIN then calls subroutine SCHOFI to solve the equations.

I.2 Subroutine SCHOFI

SCHOFI's function is to read data pertaining to pollution discharges, component concentrations along the estuary and other physical data of the estuary. It then goes into a loop which solves the equations for each day modeled by use of other subroutines.

I.3 Subroutine STORE

This is the subroutine which does the actual reading of data for pollutants depending on the type being read in and then returns these values to SCHOFI to be used in the solution.

I.4 Subroutine OUTPUT

Subroutine OUTPUT is used to print the output for all days modeled whether printed output or keypunched cards. It also checks to see if a comparison to actual data is wanted for verification purposes and whether or not plotted output is required.

I.5 Subroutine UPDATE

This subroutine updates parameters that may vary with time for beginning of solution, minute to minute, or day by day depending on the user. It also sets up boundary conditions for the components.

Here is where the sewage discharge rate is calculated and can be altered to fit the existing conditions. Also equations for sunlight intensity and temperature are found here and can be changed to suit the estuary being modeled.

I.6 Subroutine BLOCK DATA

BLOCK DATA is used to initialize some of the variables in the model and to give values to some of the rate constants. Here is where these rate constants may be changed to better fit the conditions. Also the default values for the components not in the model are provided here so that the user need not specify his own. It should be noted that many of the values provided pertain to the James River Estuary and can be changed, if need be, for use with other estuaries.

I.7 Subroutine RKMI

This is the actual subroutine which solves the simultaneous differential equations by use of the Runge-Kutta-Merson technique. It intergrates the partial derivatives of concentration with respect to time $\partial C/\partial t$ by use of subroutines UPDATE and FUNCT.

I.8 Subroutine FUNCT

The partial derivatives of the concentration with respect to time are evaluated here as well as the growth, death, predation and respiration rates of the biological components. The recycling of components and the reareation rates, are evaluated here.

I.9 Subroutine FIRST

This subroutine numerically evaluates the first derivatives of the components with respect to position through the use of sixth order difference equations and the endpoints are evaluated by second and fourth order difference equations.

I.10 Subroutine AREA

Subroutine AREA calculates the cross-sectional area of the estuary as a function of position, tidal phase and day of year. This equation uses a sine wave relation to describe the phase lag and may be changed if the user has a better relationship.

I.11 Subroutine VELOC

VELOC is the subroutine which calculates fluid velocity as a function of freshwater flow rate, position, cross-sectional area, and tidal phase. It also uses a sine wave function and may be changed if necessary.

I.12 Subroutine VOLUM

The volumetric flow rate as a function of freshwater flow rate, position and tidal phase is calculated and may be modified if the user has a better method.

I.13 Subroutine INTP

This subroutine uses interpolation to evaluate input data at the grid points and to determine the value of component concentrations at points other than the grid points.

I.14 Subroutine BILDAX

This subroutine plots the axis for plotting the concentrations with two marks at each given interval or at one inch intervals, puts numbers at tic mark and labels the axis. It uses subroutines Plot Number and Symbol which are internal to the system. Each computing facility must be checked to see if they are available.

I.15 Subroutine CONF

This subroutine calculates confidence intervals about each point using the gamma distribution if alpha is less than 50, and the normal distribution if alpha is greater than 50.

I.16 Subroutine IGAMMA

This subroutine calculates the incomplete gamma function for given values of x in the form

$$P(x<X) = \frac{1}{\Gamma(\alpha)} \int_0^x e^{-t} \, t^{\alpha-1} \quad .$$

APPENDIX II

DESCRIPTION OF VARIABLES

Variable	Definition
A(J)	Cross-sectional area (mi^2)
A0	Coefficient in expression for maximum tidal velocity
A1	Coefficient of first order term in expression for maximum tidal velocity
A2	Coefficient of second order term in expression for maximum tidal velocity
A3	Coefficient of third order term in expression for maximum tidal velocity
A4	Coefficient of fourth order term in expression for maximum tidal velocity
ABSE	Absolute allowable error of integration through time (ppm)
ABSX	Absolute value of the right hand side of the diffusion equation other than the source and sink terms
ACC	Defines the limit on interpolation error
ADJ	Adjustment of sewage input with time
AREA	Subroutine to calculate X-area
AVE	Average yearly water temperature (°C)
BA	Bacteria growth rate coefficient at 20°C/day

Variable	Definition
BC(J)	Boundary conditions for:
J=1	Algae (ppm)
=2	Protozoa (ppm)
=3	Zooplankton (ppm)
=4	Higher Predator (ppm)
=5	Bacteria (ppm)
BCN	Boundary condition for organic nitrogen (ppm)
C(I,IA,J)	Concentration of:
I=1	Organic carbon
=2	Carbon dioxide
=3	Organic nitrogen
=4	Ammonia
=5	Nitrite and nitrate
=6	Ortho-phosphate
=7	Oxygen deficit
=8	Algae
=9	Protozoa
=10	Zooplankton
=11	Higher Predator
=12	Bacteria
C28	Fraction of algae that is carbon
C29	Fraction of protozoa that is carbon
C58	Fraction of algae that is nitrogen

Variable	Definition
C59	Fraction of protozoa that is nitrogen
C68	Fraction of algae that is phosphorus
C69	Fraction of protozoa that is phosphorus
C210	Fraction of zooplankton that is carbon
C211	Fraction of higher predator that is carbon
C212	Fraction of bacteria that is carbon
C510	Fraction of zooplankton that is nitrogen
C511	Fraction of higher predator that is nitrogen
C610	Fraction of zooplankton that is phosphorus
C611	Fraction of bacteria that is phosphorus
CK3	Fraction of the organic nutrients recycled
CL	Factor adjusting pollution discharge
CONC(I,J)	Name of components:
I=1	Organic carbon
=2	Carbon dioxide
=3	Organic nitrogen
=4	Ammonia
=5	Nitrite and nitrate
=6	Phosphorus
=7	Oxygen deficit
=8	Algae
=9	Protozoa
=10	Zooplankton

Variable	Definition
=11	Higher predator
=12	Bacteria
CONF	Subroutine to calculate confidence intervals at each point
CSAT	CO_2 saturation concentration (ppm)
D1	Light extinction coefficient due to causes other than self-shading 1/ft
D2	Coefficient of the correlation for self-shading
DADT(J)	Partial derivative of X-area with respect to time
DCDT(I,IA,J)	Partial derivative of concentration with respect to time
DCDX(I,J)	Array of values equal to the right hand side of the diffusion equation other than the source and sink terms
DEATH1(I)	The death rate supplied by the program for the five biological components which are selected not to be in the model
DEATH2(I)	The death rate supplied by the user for the five biological components which are selected not to be in the model
DEF1(I)	The default value supplied by the program for the components not in the model, in the form of parts per million, for the twelve components

Variable	Definition
DEF2(I)	The default values supplied by the user for the components not in the model in the form of parts per million for the twelve components
DEV	Seasonal temperature deviation (°C)
DIE(J)	Death rate coefficient for:
J=1	Algae (1/hr)
=2	Protozoa (1/hr)
=3	Zooplankton (1/hr)
=4	Higher predator (1/hr)
=5	Bacteria (1/hr)
DIFF	In verification, difference between an observed and predicted concentration (ppm)
D02	Time step for integration DT/2
D03	Time step for integration DT/3
D015	Time step for integration DT/15
DT	Time step used in integration (hr)
DX	Length of Each segment (mile)
DX12	DX · 12; variable used for integration of partial derivatives with respect to position
DX60	DX · 60; variable used for integration of partial derivatives with respect to position

Variable	Definition
DY(J)	Appears in subroutine First only - vector of first derivatives of the dependent variable with respect to the independent variable (DY/DX)
DZS	Step in time (hr)
E	Variable which adjusts oxygen reaeration rate constant as a function of temperature
EA(J)	Array containing the product of the diffusion coefficient and the X-area
EADJ	Adjustment to the diffusion coefficient dependent upon grid spacing, 1/DX
ERRSET	Subroutine built into the system which is part of the extended error message facility
F1	Proportion of organic carbon in land runoff
F3	Proportion of organic nitrogen in land runoff
F5	Proportion of nitrite and nitrate in land runoff
F6	Proportion of ortho-phosphate in land runoff
FAZ	Moon phase on initial time and day in computer operation
FIRST	Subroutine to integrate first partial derivative with respect to position

Variable	Definition
FRA(J)	Fraction of following components in sewage discharge:
J=1	Organic carbon
=2	Carbon dioxide
=3	Organic nitrogen
=4	Ammonia
=5	Nitrite and nitrate
=6	Phosphorus
FUNCT	Subroutine to estimate partial derivative of component concentrations with respect to time
FX(J)	Used in interpolation to determine values at grid points, printout points and verification points
FXX	Resulting interpolated value
FY	Intermediate interpolated value
G0	Coefficient in expression for tidal lag
G1	Coefficient of first order term in expression for tidal lag
G2	Coefficient of second order term in expression for tidal lag
GRO(J)	Constant in temperature expression for growth rate coefficient of:
J=1	Algae $(°C\text{-}hr)^{-1}$
=2	Protozoa $(°C\text{-}hr)^{-1}$

Variable	Definition
=3	Zooplankton $(°C\text{-}hr)^{-1}$
=4	Higher predator $(°C\text{-}hr)^{-1}$
GRO8	Growth rate coefficient for algae (hr^{-1})
GRO9	Growth rate coefficient for protozoa (hr^{-1})
GRO10	Growth rate coefficient for zooplankton (hr^{-1})
GRO11	Growth rate coefficient for higher predator (hr^{-1})
GRO12	Growth rate coefficient for bacteria (hr^{-1})
GROW1(I)	The growth rate supplied by the program for the five biological components which are selected not to be in the model
GROW2(I)	The growth rate supplied by the user for the five biological components which are selected not to be in the model
IA	Index to indicate which one of the steps in the integration the concentrations are for
IC	Maximum number of components modeled
ICON	Logical variable to decide if confidence intervals are to be calculated
IDAY	Day of year simulated

Variable	Definition
IER	Variable exponent in expression to adjust time step depending upon integration accuracy
IGAMMA	Subroutine to find values of the incomplete gamma function
III	Number of days from the present to the next day of computer printout
IK	Grid point at which STP discharges enter. For verification purposes, it is the component number.
ILIST	Logic variable to determine if the Namelist option is exercised
IND	Number of components in the model
INDEF	Logical variable to determine which type of default values will be used for the components not in the model (1=concentration in ppm, 2=conversion rate constants)
INDEX(I)	Logical variable to determine whether or not the i^{th} component is in the model and which default value it has (0=in the model; 1=not in model, default is in ppm; 2=not in model, default is in rate con stants)

Variable	Definition
INRUN	Logical variable to determine whether the program is being run initially to reach a steady stat (=1) or whether it it is a regular run (=2)
INTP	Subroutine to interpolate variable value
IPDAY	Day of printout
IPOS	Grid point nearest actual observed data
IPRI	Logic variable to determine if punched output is required
IPT	Indicates the form of some input data
ITYPE	Defines input as a parameter or component
IWK2	Adjusts reaeration rate as a function of wind velocity
IZDAY	Initial day of year modeled
JA	Index to determine step of integration between t and t+Dt
JDAY	Day of year observed data available for verification
K(I,J)	Utility array
I=1	Diffusion coefficient (mi^2/hr)
=2	Benthal demand ($lb/mi^3 \cdot hr$)
=3	Average X-area (mi^2)
=4	Time variable pollution discharge rate ($lb/mi^3 \cdot hr$)
=5	Oxygen reaeration rate constant (hr^{-1})

Variable	Definition
=6	Land runoff (lb/mi^3·hr)
=7	Predation rate on algae (lb/mi^3·hr)
=8	Predation rate on protozoa (lb/mi^3·hr)
=9	Predation rate on zooplankton (lb/mi^3·hr)
=10	Predation rate on higher predator (lb/mi^3·hr)
=11	Natural death rate for algae (lb/mi^3·hr)
=12	Natural death for protozoa (lb/mi^3·hr)
=13	Natural death rate for zooplankton (lb/mi^3·hr)
=14	Natural death rate for higher predator (lb/mi^3·hr)
=15	Recycle of organic carbon (lb/mi^3·hr)
=16	Recycle of inorganic carbon (lb/mi^3·hr)
=17	Recycle of organic nitrogen (lb/mi^3·hr)
=18	Recycle of organic phosphorus (lb/mi^3·hr)
=19	Growth rate for algae (lb/mi^3·hr)
=20	Growth rate for protozoa (lb/mi^3·hr)
=21	Growth rate for zooplankton (lb/mi^3·hr)
=22	Growth rate for higher predator (lb/mi^3·hr)
=23	Respiration rate for algae (lb/mi^3·hr)
=24	Respiration rate for protozoa (lb/mi^3·hr)
=25	Respiration rate for zooplankton (lb/mi^3·hr)

Variable	Definition
=26	Respiration rate for higher predator ($lb/mi^3 \cdot hr$)
=27	Total ecosystem respiration rate ($lb/mi^3 \cdot hr$)
=28	Average pollution discharge rate (lb/day)
=29	Inorganic carbon reaeration rate (hr^{-1})
=30	Estuary depth, hydraulic radius (ft)
=31	Tidal phase
=32	Predation rate on bacteria ($lb/mi^3 \cdot hr$)
=33	Natural death rate of bacteria ($lb/mi^3 \cdot hr$)
=34	Respiration rate for bacteria ($lb/mi^3 \cdot hr$)
=35	Growth rate for bacteria ($lb/mi^3 \cdot hr$)
=36	Maximum fractional deviation in X-area from the average at high or low tide
=37	Maximum tidal velocity (mi/hr)
=38	Tidal lag (hr)
K2	Indicates which reaeration equation to use
K2X	Constant in expression for oxygen reaeration coefficient (day^{-1})
K11	Organic carbon utilization rate coefficient (hr^{-1})
K12	Rate coefficient of the conversion of organic nitrogen to ammonia (hr^{-1})
K36	Rate coefficient for the utilization of ortho-phosphate (hr^{-1})

Variable	Definition
KM8	Michaelis-Menten or half saturation concentration of substrate for algae (lb/mi^3)
KM9	Michaelis-Menten or half saturation concentration of substrate for protozoa (lb/mi^3)
KM10	Michaelis-Menten of half saturation concentration of substrate for zooplankton (lb/mi^3)
KM11	Michaelis-Menten or half saturation concentration of substrate for higher predator (lb/mi^3)
KM12	Michaelis-Menten or half saturation concentration of substrate for bacteria (lb/mi^3)
KP	Logic variable to determine if punched output requested on final day of computer operation
LIM	Number of days the model is run to reach steady state
LIST1	Name of the Namelist
LOG	Logic variable determining regular output (=1) or a comparison with observed data (=0)

Variable	Definition
LOGUP	Logic variable to indicate type of update:
=1	for initial definitions before solution begins
=2	for daily update
=3	for minute to minute update
MDAY	Final day of year of computer operation
NC	Parameter or pollutant number in array K(NC,J)
NCOEF	Number of estuary segments at which volumetric freshwater flow rate will change
ND	Determines positions or printed concentration output
NDATA	Total number of data values used in verification
NHSEC	Number of .01 seconds between printout
NM	Number of grid points
NN	Number of estuary sections
NO2NO3	Boundary condition of nitrite and nitrate (ppm)
NPLOT	Logical variable to determine which type of axis is to be used for each day (0=axis determined by values of concentration for each day modeled, 1=axis determined by the axis used on first day and then

Variable	Definition
	used for successive days, 2=user to supply axis to be used for all days.
NV(J)	Array of the component used in verification
NVIK	Specifies which component is being compared with the observed data
OSAT, O2SAT	Saturation concentration of oxygen, as a function of temperature (lb/mi^3 or ppm)
PHASE	Tidal phase
POS	Position at which observed data taken (mile)
POSIT	Position at which parameter or pollutant values are known (mile)
PRD1(I)	The predation rate supplied by the program for the five biological components which are selected not to be in the model
PRD2(I)	The predation rate supplied by the user for the five biological components which are selected not to be in the model
PRED(J)	Constant of linear temperature expression for the predation rate coefficient on:
J=1	Algae (1/°C·ppm·hr)
=2	Protozoa (1/°C·ppm·hr)
=3	Zooplankton (1/°C·ppm·hr)

Variable	Definition
=4	Higher predator $(1/^\circ\mathrm{C}\cdot\mathrm{ppm}\cdot\mathrm{hr})$
=5	Bacteria $(1/^\circ\mathrm{C}\cdot\mathrm{ppm}\cdot\mathrm{hr})$
PRED8	Algae predation rate coefficient $(\mathrm{lb/mi}^3\cdot\mathrm{hr})^{-1}$
PRED9	Protozoa predation rate coefficient $(\mathrm{lb/mi}^3\cdot\mathrm{hr})^{-1}$
PRED10	Zooplankton predation rate coefficient $(\mathrm{lb/mi}^3\cdot\mathrm{hr})^{-1}$
PRED11	Higher predator predation rate coefficient (hr^{-1})
PRED12	Bacteria predation rate coefficient $(\mathrm{lb/mi}^3\cdot\mathrm{hr})^{-1}$
PRI	Number of printouts per tidal cycle
PRIE	Maximum allowable error between specified and actual printout time
Q	Volumetric flow rate (cfs)
R	Factor multiplied to maximum growth rate of algae due to non-optimum sunlight intensity
RELE	Allowable relative error of integration through time
RES(J)	Constant in the temperature expression for:
J=1	Algae respiration rate coefficient $(^\circ\mathrm{C}\cdot\mathrm{hr})^{-1}$

Variable	Definition
=2	Protozoa respiration rate coefficient $(^\circ C \cdot hr)^{-1}$
=3	Zooplankton respiration rate coefficient $(^\circ C \cdot hr)^{-1}$
=4	Higher predator respiration rate coefficient $(^\circ C \cdot hr)^{-1}$
=5	Bacteria respiration rate coefficient $(^\circ C \cdot hr)^{-1}$
RES1(I)	The respiration rate supplied by the program for the five biological components which are selected not to be in the model
RES2(I)	The respiration rate supplied by the user for the five biological components which are selected not to be in the model
RESP8	Respiration rate coefficient for algae (hr^{-1})
RESP9	Respiration rate coefficient for protozoa (hr^{-1})
RESP10	Respiration rate coefficient for zooplankton (hr^{-1})
RESP11	Respiration rate coefficient for higher predator (hr^{-1})
RESP12	Respiration rate coefficient for bacteria (hr^{-1})

Variable	Definition
RKMI	Subroutine to evaluate partial derivative of component concentrations with respect to time
RR	Ratio of sunlight intensity to optimum sunlight intensity
S	Conversion factor for ppm to lb/mi^3; $1ppm=9.19\times10^6$ lb/mi^3
SCALF(I)	The scale factor which the user supplies to build the axis which the components are to be plotted on for all days modeled (in units per inch)
SS	Sum of squares of error in verification
STORE	Subroutine to read and store data
SUM	Sum or errors in verification
SUN	Daily sunlight totals (langleys)
SUNMAX	Maximum instantaneous sunlight intensity
SUNSAT	Optimum sunlight intensity
T	Real time of model operation (hr)
TDATA	Time at which actual data was taken
TEMP	Water temperature (°C)
TI	Initial time of day
TIDE	Time of high tide on first day of model operation. (hr)
TITLE(J)	Array of names of the parameter read as input data

Variable	Definition
TIMON, TIMECK	Internal system subroutines to determine CPU time for a certain amount of processing
TIC	Boundary condition for inorganic carbon (ppm)
TM	Final time of final day (hr)
TMAX	Maximum number of computer minutes allotted to run
TMIN	Machine time use
TON	Boundary condition for organic nitrogen (ppm)
TOC	Boundary condition for organic carbon (ppm)
TPO4	Boundary condition for ortho-phosphate (ppm)
TPR	Hour of day requiring output
TQQ	Measure of error in integration through time
UA(J)	Array of volumetric flow rates at the grid points (mi^3/hr)
UABS	Absolute value of fluid velocity (ft/sec)
UPDATE	Subroutine to update parameters
V1	The percentage of organic concentration used per hour supplied by the program when organic carbon is not in the model

Variable	Definition
V2	The percentage of organic concentration used per hour supplied by the user when organic carbon is not in the model
VAL(J)	Array of observed concentrations required in the verification (ppm)
VALUE	Input value of parameters for point source discharge or discrete constants
VELOC	Subroutine to calculate fluid velocity
VOLUM	Subroutine to calculate volumetric flow rate
WF	Period of the sine function used in expressions for point source discharge and sunlight intensity (hr^{-1})
WORK,WORK1,WORK2	Utility matrices used in integration, interpolation and reading in initial values
WT	$=2\Pi$, required for tidal and temperature expressions
XI	Upstream boundary (miles)
XK2	Variable to allocate user expression for oxygen reaeration (day^{-1})
X11	Organic carbon utilization rate coefficient at 20°C/day
X36	Ortho-phosphate utilization rate coefficient at 20°C/day

Variable	Definition
XO	Downstream boundary (miles)
XMIN(I)	Minimum value that the i^{th} component is expected to reach
XPOS(I)	Array of grid point positions (mile)
XP2	The square of XPOS
XQ	Volumetric freshwater flow rate (mi^3/hr)
XQCOEF(I)	Array to increase value of XQ by a multiplicative constant as XPOS increases
XSEC	Number of sec/real time day
XXX	Product of light extinction coefficient and water depth--needed to calculate sunlight intensity
XY	Concentration of the growth limiting substrate in the Michaelis-Menten expression
YYY	Constant in the time step determination expression
Z(I)	Array to specify at which grid points the volumetric flow rate will increase

APPENDIX III

PROGRAM LISTINGS

```
C     SOLUTION PROGRAM FOR GENERAL DETERMINISTIC MODEL.                   *
C                                                                         *
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,XO,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,DO2,DO3,DO15,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12) ,DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
C**********************************************************************************
C                                                                      *
      LIM=30
      KOUNT=0
C         THE INFORMATION IN THIS MAIN PROGRAM SEGMENT IS TO BE        *
C              SUPPLIED BY THE USER                                    *
C                                                                      *
      IC=12
C         DETERMINING APPROPRIATE EQUATION FOR REAERATION              *
C                                                                      *
```

```
C              K2 = OXYGEN REAERATION RATE                           *
C              U = TIDAL VELOCITY (FPS)                              *
C              H = DEPTH (FT)                                        *
C              D = DIFFUSIVITY OF OXYGEN IN WATER                    *
C                                                                    *
C         FOR DEPTHS OF 1 FT-30 FT AND VEL OF .5-1.6 FPS             *
C              PUT K2 = 1  TO USE THE FOLLOWING EQUATION:            *
C              K2 = 12.9*SQRT U/H**1.5                               *
C                                                                    *
C         FOR DEPTHS OF 2 FT-11 FT AND VEL OF 1.8-5.0 FPS            *
C              PUT K2 = 2  TO USE CHURCHILL'S EQUATION:              *
C              K2 = 11.6*U/H**1.67                                   *
C                                                                    *
C         FOR DEPTHS OF 4 FT-11 FT AND VEL OF .1-5.0 FPS             *
C              PUT K2 = 3  TO USE OWENS EQUATION:                    *
C              K2 = 21.6*U**0.67/H**1.85                             *
C                                                                    *
C         FOR AN ESTUARY WHICH DOES NOT FALL INTO ANY OF             *
C              THE 3 CATEGORIES ABOVE,                               *
C              PUT K2 = 4  TO USE O'CONNOR'S EQUATION INVOLVING      *
C              THE DIFFUSIVITY CONSTANT D: ( D=.000081 FT**2/HR )    *
C              K2 = (D*U)**.5/H**1.5                                 *
C                                                                    *
C         TO USE OWN EXPRESSION FOR REAERATION,                      *
C              PUT K2 = 0                                            *
C              PUT XK2 = EXPRESSION OR NUMERICAL VALUE               *
C                                                                    *
      K2 = 1
      XK2 = 0
C
C
C                                                                    *
```

```
      READ(5,555)NN,XI,XO,TI,IDAY,TM,MDAY,PRI,IND,TIDE,FAZ,TMAX
     *,KP,IWK2
 555  FORMAT(I3,2F10.4,F5.3,I3,F5.3,I3,F5.3,I3,3F10.5,2I1)
      KK=NN+1
      READ(5,1) (INDEX(I),I=1,IC),IDEF,NPLOT,INRUN ,ICON
 1    FORMAT(16I1)
      IF(INRUN.EQ.1) GO TO 1000
      IF(IDEF.EQ.1) GO TO 100
      DO 3 I=1,IC
      IF(INDEX(I).EQ.2) READ(5,33) DEF2(I)
 33   FORMAT(12F6.4)
      IF(INDEX(I).EQ.1) GO TO 5
      IF(INDEX(I).EQ.2) GO TO 6
      GO TO 3
 5    DO 8 IK=1,KK
      DO 8 KJ=1,2
 8    C(I,KJ,IK)=DEF1(I)*S
      GO TO 3
 6    DO 10 IK=1,KK
      DO 10 KJ=1,2
  10  C(I,KJ,IK)=DEF2(I)*S
 3    CONTINUE
      GO TO 110
 100  DO 121 I=8,IC
121   IF(INDEX(I).EQ.2) READ(5,111) PRD2(I),GROW2(I),DEATH2(I),RES2(I)
111   FORMAT(4F12.5)
      IF(INDEX(1).EQ.2) READ(5,111) V2
      IF(INDEX(3).EQ.2) READ(5,111) B2
      GO TO 110
1000  CONTINUE
      KP=1
      MDAY=MDAY+LIM+1
```

```
      DO 1001 I=1,IC
      DO 1001 JA=1,KK
      DO 1001 JN=1,2
1001  C(I,JN,JA)=0
C*******************************************************************************
110   CONTINUE
      DO 999 JJJ=1,IC
      IF(INDEX(JJJ).GT.0) GO TO 999
      IF(NPLOT.EQ.2)READ(5,111) XMIN(JJJ),SCALF(JJJ)
 999  CONTINUE
      DO 990 I=1,IC
      IF(INDEX(I).GT.0) GO TO 990
      IF (ICON.EQ.1) READ(5,888) DLT(I),CCOEF(I)
888   FORMAT(2F6.4)
990   CONTINUE
      NNN=0
      NPL=0
      CALL SCHOFI
      IF (NPL .GT. 0) CALL PLOT(0.,0.,-4)
      STOP
      END
```

```
C THE ORIGINAL MAIN PROGRAM
C
      SUBROUTINE SCHOFI
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,X0,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,D02,D03,D015,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
C*******************************************************************************
C
C         NAMELIST OPTION TO MODIFY DEFAULT VALUES OR TO UPDATE ACC *
C              PARAMETERS INCLUDED IN LIST.                        *
C         ANY VARIABLE MENTIONED IN THE ABOVE COMMON STATEMENT     *
C              MAY BE INCLUDED IN THIS LIST                        *
C
C*******************************************************************************
      NAMELIST /LIST1/ABSE,RELE,BC,PRED,GRO,RES,BA,X11,X14,X36,FRA,DIE,
```

```
     1DT,F1,F3,F5,F6,PRIE,KM8,KM9,KM10,KM11,KM12,CL,BCN,IPRI,YYY,AVE,DEV
     2,K2X,D1,D2,CK3
      CALL ERRSET (207,260,-1,1,0,208)
      READ(9,LIST1,END=1005)
      WRITE(6,LIST1)
C         SUBROUTINES TIMON AND TIMECK ARE SUPPLIED ON THE SYSTEM USED *
C              FOR THIS WORK.THEY MOST BE DUMMIED OUT IF NOT AVAILABLE.*
      CALL TIMON
C         READING NAMES FOR THE TWELVE POLLUTION COMPONENTS            *
      READ(5,556)A0,A1,A2,A3,A4
 556  FORMAT(5F12.10)
      READ(5,557)G0,G1,G2
 557  FORMAT(3F12.10)
      READ(5,558)NCOEF
 558  FORMAT(I2)
      DO 339 KM=1,10
 339  READ(5,559)Z(KM),XQCOEF(KM)
 559  FORMAT(2F10.6)
 1005 DO 10 I=1,12
   10 READ(5,17,END=999) (CONC(I,J),J=1,10)
   17 FORMAT(10A4)
      PRI=12.425/PRI
      IZDAY=IDAY
      T=TI
      TPR=TI
      IPDAY=IDAY
      DX=(XO-XI)/NN
      NM=NN+1
      WRITE(6,18)NM,XI,XO,TI,IDAY,TM,MDAY,PRI,IND,TIDE,FAZ
   18 FORMAT('1',19X,'NO OF GRID POINTS',14X,I5/          20X,'UPSTREAM
     1BOUNDARY,MILES',10X,F5.1/          20X,'DOWNSTREAM BOUNDARY',14X,
     2F5.1/          20X,'INITIAL TIME OF DAY',14X,F5.1/
```

```
     320X,'INITIAL DAY OF YEAR',12X,I5/20X,'FINAL TIME OF DAY',16X,F5.1/
     420X,'FINAL DAY',22X,I5/20X,'PRINT INTERVAL,HOURS',14X,F5.2/
     520X,'NO OF SIMU DIFF EQUATIONS',6X,I5/          20X,'TIME OF HIGH
     6TIDE ON FIRST DAY',4X,F5.2/20X,'MOON PHASE AT BEGINNING',11X,F5.2)
C*******************************************************************************
C     FORMING POSITION VECTOR (XPOS), MAXIMUM TIDAL VELOCITY VECTOR    *
C        (K(37,*)), AND TIDAL PHASE LAG VECTOR (K(38,*))               *
      DO 4 J=1,NM
      XPOS(J)=XI+FLOAT(J-1)*DX
      XP2=XPOS(J)*XPOS(J)
C
      K(37,J) = A0+A1*XPOS(J )+A2*XP2+A3*XP2*XPOS(J)+A4*XP2*XP2
C
      K(38,J) = G0+G1*XPOS(J)+G2*XP2
C
C*******************************************************************************
    4 CONTINUE
C     READING CONDITIONS AND SOME PARAMETER VALUES A HEADER            *
C         CARD MUST BE READ BEFORE EACH NEW SET OF DATA.THE HEAD CARD  *
C         CONTAIN VALUES FOR IPT,NC,ITYPE,AND TITLE.IPT INDICATES THE  *
C         FORM OF THE DATA,NC WHICH PARAMETER OR POLLUTANT WILL BE DES *
C         CRIBED AND ITYPE INDICATES IF IT IS A PARAMETER OR A POLLUT. *
C     IPT=; 1 MEANS STP DISCHARGE, 2 MEANS DISCRETE CONSTANTS,>2 CONT.*
C         INUOUS DATA.                                                 *
C     NC= ;I MEANS THE ITH PARAMETER OR POLLUTANT IS BEING DEFINED     *
C     ITYPE= ; 1 MEANS PARAMETER, 2 MEANS POLLUTANT ,AND 0 MEANS INPUT *
C         COMPLETED.                                                   *
    1 READ(5,6,END=999) IPT,NC,ITYPE,TITLE
    6 FORMAT(3I5,5X,10A4)
      IF(ITYPE.LT.1) GO TO 7
      CALL  STORE(IPT,XI,DX,NM,WORK,WORK1,WORK2,XPOS,CL,K)
      GO TO(8,9),ITYPE
```

```
    8 DO 12 J=1,NM
   12 K(NC,J)=WORK(J)
    9 IF(IPT.LE.2) WRITE(6,3) TITLE,(WORK(J),J=1,NM)
    3 FORMAT(///  20X,10A4 //(' ',10F12.2))
 1002 FORMAT(///20X,10A4//(' ',10F12.4))
      IF(IPT.GT.2) WRITE(6,1002) TITLE,(WORK(J),J=1,NM)
      IF(ITYPE.EQ.1) GO TO 1
      DO 13 J=1,NM
   13 C(NC,1,J)=WORK(J)*S
      GO TO 1
C                                                                   *
C              CARDS READ ON UNIT 8 SERVE ONLY MODEL VERIFICATION   *
C              PURPOSES                                             *
C                                                                   *
C     READING FIRST OBSERVED DATA CARD.                                *
C                                                                   *
    7 READ(8,11,END=999) JDAY,TDATA,POS,(NV(J),VAL(J),J=1,6)
   11 FORMAT(I5,F5.0,F5.2,5X,6(I2,F8.3))
C     READING CONDITIONS FOR FIRST DAY IF(IPR1.EQ.1) PUNCHED CARD OUT  *
C         PUT IS REQUESTED.IF(ILIST.NE.0) PARAMETER UPDATE IS REQUESTED*
C         THROUGH NAMELIST OPTION.                                     *
      READ(5,14,END=999) TEMP,Q,SUN,IPRI,ILIST,IPLOT
  14  FORMAT(3F10.4,20X,3I5)
      CALL AREA(IDAY,IZDAY,T,TIDE,FAZ,NM,XAREA,K,WT,A)
      OSAT=(14.652-0.4102*TEMP+0.00799*TEMP*TEMP-0.77774*TEMP*TEMP/10000
     ..0*TEMP)*S
      CALL UPDATE
      CALL FUNCT(1)
      GO TO 39
C     MAIN PROGRAM LOOP.                                               *
   36 DO2=DT/2.0
      DO3=DT/3.0
```

```
      DO15=DT/15.0
C     CHECKING FOR FINISHING TIME.                                      *
      IF(INRUN.EQ.1.AND.KOUNT.GT.LIM) GO TO 1000
      IF(T.GE.TM.AND.IDAY.GT.MDAY) GO TO 1000
C     CHECKING FOR MACHINE TIME ALLOCATION.                             *
      IF(TMIN.GT.TMAX) GO TO 1000
C     CALLING INTEGRATION SUBROUTINE.                                   *
      CALL RKMI(IER)
C     ADJUSTING TIME STEP(DT).                                          *
      DT=DT*YYY**IER
C     CHECKING FOR ACCURACY.                                            *
   40 IF(IER.LT.3) GO TO 41
   42 T=T-DZS
      CALL UPDATE
      CALL FUNCT(1)
      GO TO 36
C     CHECKING FOR UPPER LIMITS ON DT.                                  *
   41 IF(DT.GT.PRI) DT=PRI
      IF(DT.GT.1.4) DT=1.4
      IA=3-IA
   39 DZS=DT
      CALL OUTPUT
      DT=DZS
C     CHECKING FOR NAMELIST OPTION FOR PARAMETER UPDATE                 *
      IF(ILIST.EQ.0) GO TO 36
      READ(9,LIST1)
      WRITE(6,LIST1)
      ILIST=0
      GO TO 36
C     CHECKING FOR PUNCHED OUTPUT OF FINAL CONDITIONS                   *
 1000 IF(KP.EQ.0) GO TO 994
      DO 998 I=1,IC
```

```
      IF (INDEX(I).GE.1) GO TO 998
      DO 996 J=1,NM
  996 C(I,IA,J)=C(I,IA,J)/S
      WRITE(7,997) NM,I,(CONC(I,J),J=1,10),(C(I,IA,J),XPOS(J),J=1,NM)
998   CONTINUE
  997 FORMAT(2I5,'    2',5X,10A4/(8F10.4))
  994 WRITE(6,995) SUM,SS,NDATA,IDAY,T
  995 FORMAT('1'///' SUM OF ERRORS=',F10.2,'   SUMSQ ERRORS=',F10.2,'
     . NO OF DATA VALUES=',I5//////'   DAY=',I5,'    TIME=',F7.3)
 999  RETURN
      END
      SUBROUTINE STORE(IPT,XI,DX,NM,WORK,WORK1,WORK2,XPOS,CL,R)
```

```
C     SUBROUTINE TO READ AND STORE INITIAL INPUT DATA FOR CONCENTR-     *
C         ATIONS AND PARAMETERS.                                        *
C     INPUT FORMAT IF IPT=                                              *
C     1 DATA IS IN THE FORM DISCHARGE RATE(IN LBS/DAY)AND LOCATION OF   *
C         DISCHARGE(IN MILES) WITH A (2E20.5) FORMAT.MORE THAN ONE STP *
C         CAN BE CONSIDERED.THE STP DATA INPUT TERMINATES WHEN A NEGAT.*
C         POSITION IS READ.                                             *
C     2 DATA IS IN FORM OF VALUE-POSITION.THE PARAMETER OR POLLUTANT IS*
C         GIVEN THE READ VALUE FOR ALL LOCATIONS BETWEEN THE PREVIOUS  *
C         POSITION(IF ANY OR XI) AND THE PRESENT POSITION.THIS TERMIN- *
C         ATES WHEN A ZERO OR NEGATIVE POSITION IS READ.FORMAT(2E20.5).*
C    72 DATA IS IN THE FORM OF VALUE AND POSITION PAIRS WITH THE NO. OF*
C         PAIRS EQUAL TO IPT.THE DATA IS TREATED AS VALUES FROM A CONT-*
C         INUOUS FUNCTION AND VALUES ARE INTERPOLATED FOR LOCATIONS    *
C         BETWEEN THE POSITIONS GIVEN.                                  *
C     WORK1 AND WORK2 ARE DIMENSIONED THE GREATER OF IPT OR NM
      DIMENSION WORK(NM),WORK1(80) ,WORK2(80) ,XPOS(NM),R(38,NM)
C     IPT=1 FOR FF,IPT=2 FOR DISCRETE CONSTANTS,AND IPT>2 FOR INTERPOLING.
      IF(IPT.GT.2) GO TO 3
      IF(IPT.EQ.2) GO TO 2
      DO 9 J=1,NM
    9 WORK(J)=0.0
    4 READ(5,1,END=999) VALUE,POSIT
    1 FORMAT(2E20.5)
      IF(POSIT.LT.XI) GO TO 999
      IK=IFIX((POSIT-XI)/DX+1.5)
      IF(IK.EQ.1) IK=2
      IF(IK.GE.NM) IK=NM-1
      WORK(IK)=WORK(IK)+VALUE*CL/(DX*24.0*R(3,IK))
      GO TO 4
    2 IK=0
    6 READ(5,1,END=999) VALUE,POSIT
```

```
      IF(POSIT.LE.XI) GO TO 999
      II=IK+1
      IK=IFIX((POSIT -XI)/DX+1.5)
      IF(IK.GT.NM) IK=NM
      DO 5 I=II,IK
    5 WORK(I)=VALUE
      GO TO 6
    3 READ(5,7,END=999) (WORK1(J),WORK2(J),J=1,IPT)
    7 FORMAT(8F10.6)
      DO 8 I=1,NM
      X=XPOS(I)
      CALL INTP(3,IPT,WORK2,WORK1,X,Y,0.001,NM)
    8 WORK(I)=Y
  999 RETURN
      END
```

```
      SUBROUTINE OUTPUT
C     SUBROUTINE TO PROVIDE SPECIFIED OUTPUT,MAKE COMPAIRISON WITH ACT-*
C         UAL DATA,KEEP TRACK OF TIME AND DAY OF SOLUTION,ETC.LOG IS A *
C         LOGIC VARIABLE WHICH INDICATES WHETHER STANDARD PRINTED OUT- *
C         PUT OR A COMPARISON TO ACTUAL DATA WILL BE REQUIRED AT THE   *
C         NEXT OUTPUT TIME.                                             *
      REAL K,K11,K14,K22,K27,K36,KM8,KM10,KM11,KM12,K12,K2X,MU,KM9,LCL
      DIMENSION HOLD(16),RLIT(15),HOLD2(16),HOLD3(16),STRING(60)
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GR08,GR09,GR010,IC,E,GR011,XO,K2X,FRA(6),DIE(5),
     *RESP12,GR012,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,DO2,DO3,DO15,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
      DATA RLIT(11)/' CON'/, RLIT(12)/'CENT'/, RLIT(13)/'RATI'/,
     *RLIT(14)/'ON-P'/, RLIT(15)/'PM  '/
C     LOG=1 MEANS REG.OUTPUT ;LOG=0 MEANS A COMPARISON WITH ACTUAL DATA.
   14 IF(ABS(TPR-T).LT.PRIE.AND.IPDAY.EQ.IDAY) GO TO 21
   22 IF(T+DZS.LT.TPR.OR.IDAY.LT.IPDAY) GO TO 999
```

```
   23 DT=TPR-T
      DO3=DT/3.
      DO2=DT/2.
      DO15=DT/15.
      CALL RKMI(IER)
      IA=3-IA
   21 IF(LOG.EQ.1) GO TO 1
C     COMPARISON TO ACTUAL DATA WILL BE MADE AND THE DIFFERENCE PUNCHED*
C         FOR DETAILED ANALYSIS                                        *
      IK=1
 2    IF (INDEX(I).GE.1) GO TO 101
      IF(NV(IK).EQ. 0.OR.IK.GT.6) GO TO 3
      NVIK=NV(IK)
      DO 4 J=1,NM
      WORK(J)=C(NVIK,IA,J)
    4 WORK2(J)=K(31,J)
      CALL INTP(5,NM,XPOS,WORK,POS,FXX,0.001,NM)
      DIFF=VAL(IK)-FXX/S
      SUM=SUM+DIFF
      SS=SS+DIFF*DIFF
      NDATA=NDATA+1
      CALL INTP(5,NM,XPOS,WORK2,POS,PHASE,0.001,NM)
      IPOS=IFIX((POS-XI)/DX+1.499)
      VELO=UA(IPOS)/A(IPOS)
      WRITE(7,5) NV(IK),DIFF,T,IDAY,PHASE,SUN,VAL(IK),VELO,POS
    5 FORMAT(I5,2F10.4,I5,5F10.4)
101   IK=IK+1
      GO TO 2
    3 READ(8,9,END=50) JDAY,TDATA,POS,(NV(J),VAL(J),J=1,6)
    9 FORMAT(I5,F5.0,F5.2,5X,6(I2,F8.2))
      IF(POS.GT.XO.OR.POS.LT.XI) GO TO 3
      GO TO 10
```

```
   50 JDAY=10000
      GO TO 10
C     STANDARD PRINTED OUTPUT .SEE OUTPUT LISTING FOLLOWING THIS PRO-  *
C            GRAM LISTING FOR DETAILS.                                 *
    1 CALL TIMECK(NHSEC)
      XSEC=FLOAT(NHSEC)/100.0/PRI*24.0
      TMIN=TMIN+FLOAT(NHSEC)/6000.0
      IF(INRUN.EQ.1.AND.KOUNT.LT.LIM) GO TO 900
      WRITE(6,7) IDAY,T,DZS,XSEC,TMIN,(XPOS(J),J=1,NM,ND)
    7 FORMAT('1'///10X,'DAY=',I5,'  TIME=',F5.2,'   DELTA T=',F5.2,'
     .MACHINE TIME=',F10.3,' SEC/REAL TIME DAY    MACHINE MIN USED=',
     *F9.2//23X,'POSITION,MILES'/(' ',16F8.3))
      WRITE(6,11) (K(31,J),J=1,NM,ND)
   11 FORMAT(/23X,'TIDAL PHASE'/(' ',16F8.3))
C     PUNCHED OUTPUT OPTION (IPRI=1).                                  *
      IF(IPRI.EQ.1) WRITE(7,16) IDAY,T
   16 FORMAT('$$$$$$*********$$$$$$',I5,F10.4)
      DO 6 I=1,IC
      IF (INDEX(I).GE.1) GO TO 6
      DO 13 J=1,NM
   13 C(I,IA,J)=C(I,IA,J)/S
      WRITE(6,8) (CONC(I,J),J=1,10),(C(I,IA,J),J=1,NM,ND)
      IF(IPRI.EQ.1) WRITE(7,15) NM,I,(CONC(I,J),J=1,10),(C(I,IA,J),
     *XPOS(J),J=1,NM)
      DO 6 J=1,NM
      C(I,IA,J)=C(I,IA,J)*S
 6    CONTINUE
      IF(ICON.EQ.1) GO TO 700
      IF (IPLOT .EQ. 0) GO TO 38
      KK=0
      NPL=NPL+1
      SCALX=XO/16.
```

```
      DO 33 I=1,IC
      IF(INDEX(I).GT.0) GO TO 33
      KK=KK+1
      L=0
      DO 34 J=1,NM,ND
      L=L+1
 34   HOLD(L)=C(I,IA,J)/S
      IF(NPL .EQ. 1 .AND.KK .EQ. 1) GO TO 36
      CALL PLOT(12.,0.,-3)
      GO TO 35
  36  CALL PLOT(3.,2.,-3)
 35   DO 37 J=1,10
 37   RLIT(J)=CONC(I,J)
      IF(NPLOT.EQ.1.AND.NNN.GE.1) GO TO 100
      IF(NPLOT.EQ.2)GO TO 100
      CALL SCALE(HOLD,16,8.,XMIN(I),SCALF(I),1)
      XMIN(I)=XMIN(I)-.2*XMIN(I)
      SCALF(I)=SCALF(I)+.2*SCALF(I)
 100  CALL BILDAX(0.,0.,0.,0.,1,SCALX,8.,' POSITION-MILES ',4,.5,1,0)
      CALL BILDAX(0.,0.,0.,XMIN(I),3,SCALF(I)/2.,8.,RLIT,15,.5,1,1)
      L=0
      DO 33 J=1,NM,ND
      RMILE=XPOS(J)*(8./XO)
      L=L+1
      IF(NPLOT.EQ.2) HOLD(L)=(HOLD(L)/SCALF(I))-(XMIN(I)/SCALF(I))
      CALL PLOT(RMILE,HOLD(L),2)
  33  CONTINUE
      GO TO 200
700   NPL=NPL+1
      KK=0
      SCALX=XO/16
      DO 200 I=1,IC
```

```
      IF(INDEX(I).GT.0) GO TO 200
      KK=KK+1
      L=0
      DO 210 J=1,NM,ND
      L=L+1
  210 HOLD(L)=C(I,IA,J)/S
      IF(NPL.EQ.1.AND.KK.EQ.1) GO TO 300
      CALL PLOT(12.,0.,-3)
      GO TO 310
  300 CALL PLOT(3.,2.,-3)
 310  DO 320 J=1,10
  320 RLIT(J)=CONC(I,J)
      LK=0
      DO 400 NI=1,NM,ND
      LK=LK+1
      MU=C(I,IA,NI)/S
      ALPHA=MU/DLT(I)
      CALL CONF(ALPHA,DLT(I),CCOEF(I),UCL,LCL,MU,GAMAIN,IFLAG)
      HOLD2(LK)=UCL
      HOLD3(LK)=LCL
  400 CONTINUE
      KL=NM/ND
      DO 500 LJ=1,KL
  500 STRING(LJ)=HOLD2(LJ)
      KKL=KL
      DO 510 NK=1,KKL
  510 STRING(KL+NK)=HOLD3(NK)
      NKK=KL
      NNK=KL+KKL
      DO 777 JL=1,NKK
  777 STRING(NNK+JL)=HOLD(JL)
      MJ=NNK+KL
```

```
      CALL SCALE(STRING,MJ,8.,XMIN(I),SCALF(I),1)
      CALL BILDAX(0.,0.,0.,0.,1,SCALX,8.,' POSITION-MILES ',4,.5,1,0)
      CALL BILDAX(0.,0.,0.,XMIN(I),3,SCALF(I)/2.,8.,RLIT,15,.5,1,1)
      DO 333 LJ=1,KL
  333 HOLD2(LJ)=STRING(LJ)
      DO 444 NK=1,KKL
  444 HOLD3(NK)=STRING(KL+NK)
      DO 555 JL=1,NKK
  555 HOLD(JL)=STRING(NNK+JL)
      L=0
      DO 600 JJ=1,NM,ND
      RMILE=XPOS(JJ)*(8./XO)
      L=L+1
  600 CALL PLOT(RMILE,HOLD(L),2)
      L=0
      DO 610 JJ=1,NM,ND
      L=L+1
      RMILE=XPOS(JJ)*(8./XO)
      IF(JJ.EQ.1) GO TO 620
  630 CALL PLOT(RMILE,HOLD2(L),2)
      GO TO 610
  620 CALL PLOT(RMILE,HOLD2(L),3)
 610  CONTINUE
      L=0
      DO 710 JJ=1,NM,ND
      L=L+1
      RMILE=XPOS(JJ)*(8./XO)
      IF(JJ.EQ.1) GO TO 720
  730 CALL PLOT(RMILE,HOLD3(L),2)
      GO TO 710
  720 CALL PLOT(RMILE,HOLD3(L),3)
  710 CONTINUE
```

```
 200  CONTINUE
      NNN=NNN+1
  38  IPRI=0
    8 FORMAT(// 20X,10A4/ (' ',16F8.3))
   15 FORMAT(2I5,'     2',5X,10A4/(8F10.4))
      WRITE(6,17) Q,TEMP,SUN
   17 FORMAT('0Q=',F10.2,'   TEMP=',F10.2,'   SUN LIGHT INTENS=',F10.2)
900   CONTINUE
      CALL TIMON
   10 III=IFIX((T+PRI)/24.0)
      TPR=T+PRI-24.0*FLOAT(III)
      IPDAY=IDAY+III
      LOG=1
C*******************************************************************************
C
C          THE FOLLOWING CARDS MAY BE OMITTED IF MODEL VERIFICATION  *
C               IS NOT INTENDED.                                     *
C
      IF(JDAY.GE.IPDAY.AND.TDATA.GE.TPR) GO TO 999
      IF(JDAY.GT.IPDAY.AND.TDATA.LT.TPR) GO TO 999
      TPR=TDATA
      IPDAY=JDAY
      LOG=0
      GO TO 14
C
C*******************************************************************************
  999 IF(T.LT.24.0) GO TO 1000
      IF(INRUN.EQ.1) GO TO 901
      IF(IDAY.EQ.MDAY)GO TO 51
      READ(5,12,END=51) TEMP,Q,SUN,IPRI,ILIST,IPLOT
   12 FORMAT(3F10.4,20X,3I5)
C     UPDATING TIME,DAY,AND CONDITIONS.                                 *
```

```
901    CONTINUE
   51 IDAY=IDAY+1
      KOUNT=KOUNT+1
      T=T-24.0
      LOGUP=2
      CALL UPDATE
 1000 RETURN
      END
```

```
      SUBROUTINE UPDATE
C     SUBROUTINE TO UPDATE ANY TIME,DAY,OR TEMPERATURE VARIABLE PARA-  *
C         METERS,FORMING FUNCTIONS,CONDITIONS,ETC.                     *
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,XO,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,DO2,DO3,DO15,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL
     *,INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
C     LOGUP IS A LOGIC VARIABLE WHICH INDICATES WHAT TYPE OF UPDATE IS *
C         TO BE EXECUTED.                                              *
C     LOGUP=1 BEFORE SOLUTION BEGINS FOR INITIAL DEFINITIONS.          *
C          =2 FOR DAILY UPDATE.                                        *
C          =3 FOR MINUTE TO MINUTE UPDATE.                             *
      GO TO(1,2,3),LOGUP
    1 DO 4 I=8,12
      DO 4 J=1,2
    4 C(I,J,1)=BC(I-7)*S
```

```
      EADJ=1.0/DX
      ND=1+(NM-1)/16
    2 SUNMAX=SUN*2.6224
C*******************************************************************************
C         IF TEMP IS NOT INCLUDED IN THE DATA THE FOLLOWING
C              EXPRESSION WILL CALCULATE ITS VALUE:                 *
C
      IF(TEMP.EQ.0.0) TEMP=AVE+DEV*SIN(WT/365.0*FLOAT(IDAY-120))
C
C         DEFAULT VALUES FOR AVE AND DEV FOUND IN BLK DATA SHOULD   *
C              BE CHECKED FOR APPROPRIATENESS FOR EACH ESTUARY      *
C              MODELED.                                             *
C
C*******************************************************************************
      PRED8=PRED(1)*TEMP/S
      PRED9=PRED(2)*TEMP/S
      PRED10=PRED(3)*TEMP/S
      PRED11=PRED(4)*TEMP
      GRO8=GRO(1)*TEMP+0.02
      GRO9=GRO(2)*TEMP
      GRO10=GRO(3)*TEMP
      GRO11=GRO(4)*TEMP
      GRO12= BA*1.050**(TEMP-20.0)/24.0
      RESP8=RES(1)*TEMP
      RESP9=RES(2)*TEMP
      RESP10=RES(3)*TEMP
      RESP11=RES(4)*TEMP
      RESP12=RES(5)*TEMP
      E=1.021**(TEMP-20.0)
      CSAT=(2550.0-43.0*TEMP)*S
C     K11= X11*1.047**(TEMP-20.0)/24.0
      K11= X11*1.010**(TEMP-20.0)/24.0
```

```
      K12=X14*1.010**(TEMP-20.0)/24.0
      K14= X14*1.188**(TEMP-20.0)/24.0
C     K36=  X36*1.084**(TEMP-20.0)/24.0/S
      K36=  X36*1.010**(TEMP-20.0)/24.0/S
      O2SAT=(14.652-0.4102*TEMP+0.00799*TEMP*TEMP-0.77774*TEMP*TEMP/1000
     .0.0*TEMP)*S
      ODIFF=O2SAT-OSAT
      DO 6 J=1,NM
    6 C(7,IA,J)=C(7,IA,J)+ODIFF
      OSAT=O2SAT
C     BOUNDARY CONDITIONS                                              *
      BOD=67.3*(1/Q**.1828)/5.4*S
      PI=0.110*Q**(1.209-1.0)/5.38*S
      TPO4=0.101*Q**(1.276-1.0)/5.38*S
      TKN=2.797*Q**(1.012-1.0)/5.38*S
      TOC=33.960*(1/Q**.035)/5.38*S
      NO2NO3=19.590*(1/Q**.220)/5.38*S
      TIC=2468.0*(1/Q**.4605)/5.4*S*0.5
      C(1,1,1)=TOC
      C(1,2,1)=TOC
      C(2,1,1)=TIC
      C(2,2,1)=TIC
      C(5,1,1)=NO2NO3
      C(5,2,1)=NO2NO3
      C(6,1,1)=TPO4
      C(6,2,1)=TPO4
      C(4,1,1)=0.025*Q**(1.37-1.0)/5.38*S
      C(4,2,1)=0.025*Q**(1.37-1.0)/5.38*S
      TON=TKN-C(4,1,1)
      C(3,1,1)=TON
      C(3,2,1)=TON
      C(7,1,1)=3.0*(1.0-Q/(10000.0+Q))*S
```

```
      C(7,2,1)=C(7,1,1)
    3 SUN=0.0
      IF(T.LT.6.0.OR.T.GT.18.0) GO TO 8
C     SUNLIGHT INTENSITY                                                *
      SX=SIN(WF*(T-6.0))
      IF(SX.LT.0.0) SX=0.0
      SUN=SUNMAX*SQRT(SX)
C     STP DISCHARGE RATE                                                *
    8 ADJ=1.0+0.4*SIN(WF*(T-16.0))
      DO 5 J=1,NM
    5 K(4,J)=ADJ*K(28,J)*K(3,J)/A(J)
      RR=SUN/SUNSAT
      RETURN
      END
```

```
      BLOCK DATA
C     SUBPROGRAM TO DEFINE DEFAULT VALUES FOR VARIABLES LISTED         *
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,X0,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,DO2,DO3,DO15,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
      DATA CK3/0.9/,KM8,KM9,KM10,KM11/0.23E07,0.66E07,9.19E06,9.19E06/,
     .C28,C29,C210,C211/.70,.83,.77,.83/,C58,C59,C510,C511/.28,.14,.18,
     ..14/,C68,C69,C610,C611/.07,.04,.05,.04/,C8,C9,C10,C11,C12/.66,.662
     .,.66,.662,.667/,C212,C512,C612/.77,.18,.05/,F1,F3,F5,F6/.7,.2,.1,.
     .0/,ABSE,RELE,LOG,S,DT,IA,LOGUP/ 500000.,0.000,1,9.19E+06,1.0,1,1/,
     .SUNSAT,WF/300.0,0.2618/,BC/5*.05/,PRED/3*0.0002,0.0001,.0/,GRO/.00
     .40, 0.004, 0.004, 0.002,0.0/,RES/2*0.0002,0.0002,0.0001,.0001/,BA,
     .X11,X14,X36/0.33,0.230,0.068,0.0225/,FRA/0.40,0.05,0.40,0.05,0.0
     .,0.10/,DIE/0.010,2*0.0050,0.0020,0.005/,KM12,PRIE,CL/9.19E+06,0.05
     .,1.0/,C/1440*9.19E06,3360*0.0/,DCDX/2400*0.0/,K/7600*0.0/,SS,SUM,
```

```
.BCN,NDATA/0.0,0.0,0.1,0/,TMIN/0.0/,IPRI/0/,YYY/.970/,AVE,
.DEV/16.,14./,ND/2/,WT/6.283185/,K2X/12.9/,D1,D2/1.0,0.05/,
.XAREA/2400*0.0/,DCDT/12000*0.0/,DEF1/3.56,3.106,.485,.573,.426,
..514,2.239,.710,.036,.026,.027,.047/
 END
```

```
      SUBROUTINE RKMI( J1)
C     SUBROUTINE TO INTEGRATE THROUGH TIME.                             *
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,X0,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C28,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,D02,D03,D015,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
      TQQ=0.0
      JA=3-IA
      LOGUP=3
C     ESTIMATING C AT  T&DT/3 USING C AND DCDT AT T                     *
      DO 1  I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 1
      DO 1  J=2,NM
      C(I,JA,J)=C(I,IA,J)+D03*DCDT(I, 1,J)
      IF(C(I,JA,J).LE.0.0) C(I,JA,J)=0.0
    1 CONTINUE
```

```
      T=T+DO3
      CALL UPDATE
      CALL FUNCT(2)
C     UPDATING ESTIMATE OF C AT T&DT/3 USING C AND DCDT AT T&DT/3      *
      DO 2 I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 2
      DO 2 J=2,NM
      C(I,JA,J)=C(I,JA,J)+DO3*(DCDT(I, 2,J)-DCDT(I, 1,J))/2.0
      IF(C(I,JA,J).LE.0.0) C(I,JA,J)=0.0
    2 CONTINUE
      CALL FUNCT(3)
      DO 3 I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 3
      DO 3 J=2,NM
      C(I,JA,J)=C(I,JA,J)+DO3*(1.125*DCDT(I,3,J)-0.5*DCDT(I,2,J)-0.125*
     .        DCDT(I,1,J))
      IF(C(I,JA,J).LE.0.0) C(I,JA,J)=0.0
    3 CONTINUE
      T=T-DO3+DO2
      CALL UPDATE
      CALL FUNCT(4)
C     ESTIMATING C AT  T+DT/2 USING C AND DCDT AT T&DT/3               *
      DO 4 I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 4
      DO 4 J=2,NM
      C(I,JA,J)=C(I,JA,J)+DO3*(1.125*DCDT(I,1,J)-5.625*DCDT(I,3,J)+6.0*
     .         DCDT(I, 4,J))
      IF(C(I,JA,J).LE.0.0) C(I,JA,J)=0.0
    4 CONTINUE
      T=T+DO2
      CALL UPDATE
      CALL FUNCT(5)
```

```
C     ESTIMATING C AT  T&DT USING C AND DCDT AT T&DT/2                   *
C     CHECKING ACCURACY                                                  *
      DO 5 I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 5
      DO 5  J=2,NM
      TQ=DCDT(I,1,J)-4.5*DCDT(I,3,J)+4.0*DCDT(I,4,J)-0.5*DCDT(I,5,J)
      C(I,JA,J)=C(I,JA,J)-D03*TQ
      IF(C(I,JA,J).LE.0.0) C(I,JA,J)=0.0
      IF(I.NE.1.AND.I.NE.3) GO TO 5
      TQ=D015*TQ/(RELE*ABS(C(I,JA,J))+ABSE)
      TQ=ABS(TQ)
      IF(TQ.GT.TQQ) TQQ=TQ
    5 CONTINUE
      J1=IFIX(TQQ)
      IF(TQQ.LT.0.10) J1=-1
      IF(J1.GT.3) J1=3
      CALL FUNCT(1)
      RETURN
      END
```

```
      SUBROUTINE FUNCT(L)
C     SUBROUTINE TO ESTIMATE DCDT                                      *
      REAL K,K11,K14,K22,K27,K36,KM8,KM9,KM10,KM11,KM12,K12,K2X
      COMMON /B0/C(12,2,200),DCDT(12,5,200 ),K(38,200),WORK(200),
     *WORK1(200),WORK2(200),DCDX(12,200),EA(200 ),UA(200),XPOS(200),
     *DEV,ND,DADT(200),A(200),WT,NM,NN,DT,DX,DZS,IA,TPR,LOGUP,INLOG
     *,ABSE,RELE,TEMP,TIDE,T,SUN,Q,O2SAT,CSAT,IDAY,JDAY,TDATA,K11,
     *K14,K27,K36,GRO8,GRO9,GRO10,IC,E,GRO11,XO,K2X,FRA(6),DIE(5),
     *RESP12,GRO12,IPRI,CL,PRED8,PRED9,PRED10,PRED11,WF,RR/B1/CK3/
     *B2/KM8,KM9,KM10,KM11/B3/C2 8,C29,C210,C211/B4/C58,C59,C510,
     *C511/B5/C68,C69,C610,C611/B6/C8,C9,C10,C11,C12,KM12,S/B7/C212,
     *C512,C612/B8/RESP8,RESP9,RESP10,RESP11/B9/F1,F3,F5,F6,FAZ,
     *IZDAY,POS,CONC(12,10),NV(6),VAL(6),TITLE(10),IPDAY,LOG,OSAT,D1,
     *PRI,DO2,DO3,DO15,BC(5),PRED(5),GRO(5),RES(5),BA,X11,X14,X36,
     *SUNSAT/B10/TMIN,ILIST,PRIE,XI,BCN,SUM,SS,NDATA,YYY,EADJ,
     *XAREA(12,200),K12,AVE,D2,TI,TM,MDAY,TMAX/B11/KP,K2,A0,A1,A2,
     *A3,A4,G0,G1,G2,XK2/B12/XQCOEF(10),Z(10),NCOEF,IWK2/B13/IPLOT,NPL,
     *INDEX(12),DEF1(12),DEF2(12)
     */B14/IDEF,PRD1(12),PRD2(12),GROW1(12),GROW2(12),RES1(12),
     *RES2(12),DEATH1(12),DEATH2(12),V1,V2,B1,B2
     *,NPLOT,NNN,XMIN(12),SCALF(12),INRUN,KOUNT,LIM,IND,DLT(12)
     *,CCOEF(12),ICON
      IA=3-IA
C   CALCULATE PREDATION
    1 DO 86 J=2,NM
      IF(IDEF.EQ.1.AND.INDEX(8).GT.0) GO TO 200
      K(7,J)=PRED8*C(8,IA,J)*(C(9,IA,J)+C(10,IA,J)+C(11,IA,J))
  400 IF(IDEF.EQ.1.AND.INDEX(9).GT.0) GO TO 210
      K(8,J)=PRED9*C(9,IA,J)*C(10,IA,J)
  410 IF(IDEF.EQ.1.AND.INDEX(10).GT.0) GO TO 220
      K(9,J)=PRED10*C(10,IA,J)*C(11,IA,J)
  420 IF(IDEF.EQ.1.AND.INDEX(11).GT.0) GO TO 230
```

```
      K(10,J)=PRED11*C(11,IA,J)
  430 IF(IDEF.EQ.1.AND.INDEX(12).GT.0) GO TO 240
      K(32,J)=PRED8*C(9,IA,J)*C(12,IA,J)
      GO TO 86
  200 IF(INDEX(8).EQ.1) K(7,J)=PRD1(8)
      IF(INDEX(8).EQ.2) K(7,J)=PRD2(8)
      GO TO 400
  210 IF(INDEX(9).EQ.1) K(8,J)=PRD1(9)
      IF(INDEX(9).EQ.2) K(8,J)=PRD2(9)
      GO TO 410
  220 IF(INDEX(10).EQ.1) K(9,J)=PRD1(10)
      IF(INDEX(10).EQ.2) K(9,J)=PRD2(10)
      GO TO 420
 230  IF(INDEX(11).EQ.1) K(10,J)=PRD1(11)
      IF(INDEX(11).EQ.2) K(10,J)=PRD2(11)
      GO TO 430
  240 IF(INDEX(12).EQ.1) K(32,J)=PRD1(12)
      IF(INDEX(12).EQ.2) K(32,J)=PRD2(12)
   86 CONTINUE
C    CALCULATE GROWTH
      DO 84 J=2,NM
      IF(IDEF.EQ.1.AND.INDEX(8).GT.0) GO TO 300
      XXX=K(30,J)*(D1+D2/C8*C(8,IA,J)/S)
      R=(EXP(-RR*EXP(-XXX))-EXP(-RR))/XXX*2.718
      K(19,J)=GRO8*R*(C(5,IA,J)/(KM8+C(5,IA,J)))*C(8,IA,J)
  500 IF(IDEF.EQ.1.AND.INDEX(9).GT.0) GO TO 310
      XY=PRED8*C(9,IA,J)*(C(8,IA,J)+C(12,IA,J))
      K(20,J)=GRO9*(XY/(KM9+XY))*C(9,IA,J)
  510 IF(IDEF.EQ.1.AND.INDEX(10).GT.0) GO TO 320
      XY=(PRED8*C(8,IA,J)+PRED9*C(9,IA,J))*C(10,IA,J)
      K(21,J)=GRO10*(XY/(KM10+XY))*C(10,IA,J)
  520 IF(IDEF.EQ.1.AND.INDEX(11).GT.0) GO TO 330
```

```
      XY=(PRED8*C(8,IA,J)+PRED10*C(10,IA,J))*C(11,IA,J)
      K(22,J)=GRO11*(XY/(KM11+XY))*C(11,IA,J)
  530 IF(IDEF.EQ.1.AND.INDEX(12).GT.0) GO TO 340
      K(35,J)=GRO12*C(4,IA,J)/(KM12+C(4,IA,J))*C(12,IA,J)
      GO TO 84
  300 IF(INDEX(8).EQ.1) K(19,J)=GROW1(8)
      IF(INDEX(8).EQ.2) K(19,J)=GROW2(8)
      GO TO 500
  310 IF(INDEX(9).EQ.1) K(20,J)=GROW1(9)
      IF(INDEX(9).EQ.2) K(20,J)=GROW2(9)
      GO TO 510
  320 IF(INDEX(10).EQ.1) K(21,J)=GROW1(10)
      IF(INDEX(10).EQ.2) K(21,J)=GROW2(10)
      GO TO 520
  330 IF(INDEX(11).EQ.1) K(22,J)=GROW1(11)
      IF(INDEX(11).EQ.2) K(22,J)=GROW2(11)
      GO TO 530
  340 IF(INDEX(12).EQ.1) K(35,J)=GROW1(12)
      IF(INDEX(12).EQ.2) K(35,J)=GROW2(12)
   84 CONTINUE
C     CALCULATE RESPIRATION
      DO 85 J=2,NM
      IF(IDEF.EQ.1.AND.INDEX(8).GT.0) GO TO 600
      K(23,J)=RESP8*C(8,IA,J)
  700 IF(IDEF.EQ.1.AND.INDEX(9).GT.0) GO TO 610
      K(24,J)=RESP9*C(9,IA,J)
  710 IF(IDEF.EQ.1.AND.INDEX(10).GT.0) GO TO 620
      K(25,J)=RESP10*C(10,IA,J)
  720 IF(IDEF.EQ.1.AND.INDEX(11).GT.0) GO TO 630
      K(26,J)=RESP11*C(11,IA,J)
  730 IF(IDEF.EQ.1.AND.INDEX(12).GT.0) GO TO 640
      K(34,J)=RESP12*C(12,IA,J)
```

```
 740  K(27,J)=K(23,J)+K(24,J)+K(25,J)+K(26,J)+K(34,J)
      GO TO 85
 600  IF(INDEX(8).EQ.1) K(23,J)=RES1(8)
      IF(INDEX(8).EQ.2) K(23,J)=RES2(8)
      GO TO 700
 610  IF(INDEX(9).EQ.1) K(24,J)=RES1(9)
      IF(INDEX(9).EQ.2) K(24,J)=RES2(9)
      GO TO 710
 620  IF(INDEX(10).EQ.1) K(25,J)=RES1(10)
      IF(INDEX(10).EQ.2) K(25,J)=RES2(10)
      GO TO 720
 630  IF(INDEX(11).EQ.1) K(26,J)=RES1(11)
      IF(INDEX(11).EQ.2) K(26,J)=RES2(11)
      GO TO 730
 640  IF(INDEX(12).EQ.1) K(34,J)=RES1(12)
      IF(INDEX(12).EQ.2) K(34,J)=RES2(12)
      GO TO 740
 85   CONTINUE
C   CALCULATE DEATH
      DO 87 J=2,NM
      IF(IDEF.EQ.1.AND.INDEX(8).GT.0) GO TO 800
      K(11,J)=DIE(1)*C(8,IA,J)
 900  IF(IDEF.EQ.1.AND.INDEX(9).GT.0) GO TO 810
      K(12,J)=DIE(2)*C(9,IA,J)
 910  IF(IDEF.EQ.1.AND.INDEX(10).GT.0) GO TO 820
      K(13,J)=DIE(3)*C(10,IA,J)
 920  IF(IDEF.EQ.1.AND.INDEX(11).GT.0) GO TO 830
      K(14,J)=DIE(4)*C(11,IA,J)
 930  IF(IDEF.EQ.1.AND.INDEX(12).GT.0) GO TO 840
      K(33,J)=DIE(5)*C(12,IA,J)
      GO TO 87
 800  IF(INDEX(8).EQ.1) K(11,J)=DEATH1(8)
```

```
      IF(INDEX(8).EQ.2) K(11,J)=DEATH2(8)
      GO TO 900
  810 IF(INDEX(9).EQ.1) K(12,J)=DEATH1(9)
      IF(INDEX(9).EQ.2) K(12,J)=DEATH2(9)
      GO TO 910
  820 IF(INDEX(10).EQ.1) K(13,J)=DEATH1(10)
      IF(INDEX(10).EQ.2) K(13,J)=DEATH2(10)
      GO TO 920
  830 IF(INDEX(11).EQ.1) K(14,J)=DEATH1(11)
      IF(INDEX(11).EQ.2) K(14,J)=DEATH2(11)
      GO TO 930
  840 IF(INDEX(12).EQ.1) K(33,J)=DEATH1(12)
      IF(INDEX(12).EQ.2) K(33,J)=DEATH2(12)
   87 CONTINUE
C    CALCULATE RECYCLING
      DO 89 J=2,NM
      K(15,J)=CK3*(C28*(K(7,J)+K(11,J))+C29*(K(8,J)+K(12,J)-K(20,J)-K(24
     .        ,J))+C210*(K(9,J)+K(13,J)-K(21,J)-K(25,J))+C211*(K(10,J)+K(
     .        14,J)-K(22,J)-K(26,J))+C212*(K(32,J)+K(33,J)))
      K(16,J)=C28*K(23,J)+C29*K(24,J)+C210*K(25,J)+C211*K(26,J)+C212*
     .          K(34,J)
      K(17,J)=CK3*(C58*(K(7,J)+K(11,J)+K(23,J))+C59*(K(8,J)+K(12,J)-K(20
     .,J))+C510*(K(9,J)+K(13,J)-K(21,J))+C211*(K(10,J)+K(14,J)-K(22,J))+
     .C512*(K(32,J)+K(33,J)+K(34,J)))
   89 K(18,J)=CK3*(C68*(K(7,J)+K(11,J)+K(23,J))+C69*(K(8,J)+K(12,J)-K(20
     .,J))+C610*(K(9,J)+K(13,J)-K(21,J))+C611*(K(10,J)+K(14,J)-K(22,J))+
     .C612*(K(32,J)+K(33,J)+K(34,J)))
C     UPDATING VALUES FOR AREA,VELOCITY,VOLUMETRIC FLOW RATE AND THEN  *
C         REAERATION RATES,AND DIFFUSION COEFFICIENTS.                 *
      CALL AREA(IDAY,IZDAY,T,TIDE,FAZ,NM,XAREA,K,WT,A)
      CALL VELOC(XPOS,Q,A,WT,NM,K,UA)
      CALL VOLUM(A,NM,UA,Q,WORK,XPOS)
```

```
   40 DO 88 J=1,NM
      UABS=ABS(UA(J))*1.466
      K(1,J)=0.0036*UABS*K(30,J)*EADJ
C*******************************************************************************
C                                                                   *
C         FORMULATIONS FOR K2 OPTION                                *
      GO TO ( 10,20,30,35 ), K2
      IF (K2 .EQ. 0) K(5,J)=XK2
      GO TO 36
 10   K(5,J) = K2X*SQRT(UABS/K(30,J))/K(30,J)
      GO TO 36
 20   K(5,J) = 11.6*UABS/K(30,J)**1.67
      GO TO 36
 30   K(5,J) = 21.6*UABS**0.67/K(30,J)**1.85
      GO TO 36
 35   K(5,J) = 12.96*UABS**.5/K(30,J)**1.5
 36   K(5,J) = K(5,J)/24.0
C
C         TO COMPENSATE FOR NO WIND, 10 KNOT, 20 KNOT...60 KNOT     *
C              WINDS                                                *
      IF (IWK2 .EQ. 0) GO TO 37
      GO TO (7,8,3,4,5,6),IWK2
  7   K(5,J)=K(5,J)*1.25
      GO TO 37
  8   K(5,J)=K(5,J)*1.5
      GO TO 37
  3   K(5,J)=K(5,J)*1.75
      GO TO 37
  4   K(5,J) = K(5,J)*2.0
      GO TO 37
  5   K(5,J)=K(5,J)*2.25
      GO TO 37
```

```
  6   K(5,J)= K(5,J)*2.5
C                                                                        *
C*****************************************************************************
  37  K(29,J)=0.0003*K(5,J)
      EA(J)=K(1,J)*A(J)
   88 UA(J)=UA(J)*A(J)
      CALL FIRST(WORK,WORK2,DX,NM)
      DO 93 J=1,NM
   93 DADT(J)=-WORK2(J)
      DO 90 I=1,IC
      IF(INDEX(I).GE.1.AND.INRUN.EQ.0) GO TO 90
      DO 91 J=1,NM
   91 WORK(J)=C(I,IA,J)
      CALL FIRST(WORK,WORK2,DX,NM)
      MM=NM-2
      DO 41 J=MM,NM
   41 IF(WORK2(J).GT.0.0) WORK2(J)=0.0
      DO 92 J=1,NM
      WORK(J)=WORK2(J)*EA(J)-UA(J)*C(I,IA,J)
   92 IF(I.EQ.8) WORK(J)=0.60*WORK(J)
      CALL FIRST(WORK,WORK2,DX,NM)
      DO 90 J=2,NM
      DCDX(I,J)=(WORK2(J)-C(I,IA,J)*DADT(J))/A(J)
      ABSX=ABS(DCDX(I,J))
      IF(ABSX.GT.1.0E+08) DCDX(I,J)=DCDX(I,J)/ABSX*1.0E+08
 90   CONTINUE
      DO 94 J=2,NM
      IF(INRUN.EQ.1) GO TO 1003
C 1 ORGANIC CARBON
      IF(INDEX(1).EQ.0) DCDT(1,L,J)=DCDX(1,J)+FRA(1)*K(4,J)-K11*
     .C(1,IA,J)+K(15,J)+F1*K(6,J)
C 2 CARBON DIOXIDE
```

```
      IF(INDEX(1).GT.0.AND.INDEX(2).EQ.0) GO TO 100
101   IF(INDEX(2).EQ.0) DCDT(2,L,J)=DCDX(2,J)+FRA(2)*K(4,J)+K11*
     .C(1,IA,J)+K(29,J)*E*(CSAT
     .           -C(2,IA,J))+K(16,J)-(C28*K(19,J)+C29*K(20,J)+C210*K(21,
     .            J) +C211*K(22,J)+C212*K(35,J))
103   CONTINUE
C 3 ORGANIC NITROGEN
      IF(INDEX(3).EQ.0) DCDT(3,L,J)=DCDX(3,J)+FRA(3)*K(4,J)-K12*
     .C(3,IA,J)+K(17,J)+F3*K(6,J)
C 4 AMMONIA NITROGEN
      IF(INDEX(3).GT.0.AND.INDEX(4).EQ.0) GO TO 110
112   IF(INDEX(4).EQ.0) DCDT(4,L,J)=DCDX(4,J)+FRA(4)*K(4,J)+K12*
     .C(3,IA,J)-20.0*K(35,J)
111   CONTINUE
C 5 NITRITE AND NITRATE NITROGEN
      IF(INDEX(5).EQ.0) DCDT(5,L,J)=DCDX(5,J)+FRA(5)*K(4,J)+19.0*K(35,J)
     . -(C58*K(19,J)+
     . C59*K(20,J)+C510*K(21,J)+C511*K(22,J)+C512*K(35,J))+F5*K(6,J)
C 6 ORTHO-PHOSPHATE PHOSPHORUS
      IF(INDEX(6).EQ.0)  DCDT(6,L,J)=DCDX(6,J)+FRA(6)*K(4,J)-K36
     .*C(6,IA,J)*C(6,IA,J)+
     .            K(18,J)-(C68*K(19,J)+C69*K(20,J)+C610*K(21,J)+C611*
     .            K(22,J)+C612*K(35,J))+F6*K(6,J)
C 7 OXYGEN DEFICIT
      IF(INDEX(1).GT.0.AND.INDEX(7).EQ.0) GO TO 120
 122  IF(INDEX(7).EQ.0) DCDT(7,L,J)=DCDX(7,J)-K(5,J)*E*C(7,IA,J)+K(2,J)
     .  +2.667*(K(27,J)-C28*K(19,J)+K11*C(1,IA,J))+4.571*19.0*K(35,J)
 121  CONTINUE
C 8 ALGAE
      IF(INDEX(8).EQ.0) DCDT(8,L,J)=DCDX(8,J)+K(19,J)-K(23,J)-K(7,J)
     . -K(11,J)
C 9 PROTOZOA
```

```
      IF(INDEX(9).EQ.0) DCDT(9,L,J)=DCDX(9,J)+K(20,J)-K(8,J)-K(12,J)
C 10 ZOOPLANKTON
      IF(INDEX(10).EQ.0) DCDT(10,L,J)=DCDX(10,J)+K(21,J)-K(9,J)-K(13,J)
C 11 HIGHER PREDITORS
      IF(INDEX(11).EQ.0)DCDT(11,L,J)=DCDX(11,J)+K(22,J)-K(10,J)-K(14,J)
C 12 BACTERIA
      IF(INDEX(12).EQ.0) DCDT(12,L,J)=DCDX(12,J)+K(35,J)-K(33,J)-K(32,J)
      GO TO 94
100   IF(IDEF.EQ.0) GO TO 101
      IF(INDEX(1).EQ.1) DCDT(2,L,J)=DCDX(2,J)+FRA(2)*K(4,J)+V1
     .         +K(29,J)*E*(CSAT
     .           -C(2,IA,J))+K(16,J)-(C28*K(19,J)+C29*K(20,J)+C210*K(21,
     .            J) +C211*K(22,J)+C212*K(35,J))
      IF(INDEX(1).EQ.2) DCDT(2,L,J)=DCDX(2,J)+FRA(2)*K(4,J)+V2
     .         +K(29,J)*E*(CSAT
     .           -C(2,IA,J))+K(16,J)-(C28*K(19,J)+C29*K(20,J)+C210*K(21,
     .            J) +C211*K(22,J)+C212*K(35,J))
      GO TO 103
110   IF(IDEF.EQ.0) GO TO 112
      IF(INDEX(3).EQ.1) DCDT(4,L,J)=DCDX(4,J)+FRA(4)*K(4,J)+B1
     .         -20.0*K(35,J)
      IF(INDEX(3).EQ.2) DCDT(4,L,J)=DCDX(4,J)+FRA(4)*K(4,J)+B2
     .         -20.0*K(35,J)
      GO TO 111
120   IF(IDEF.EQ.0) GO TO 122
      IF(INDEX(1).EQ.1) DCDT(7,L,J)=DCDX(7,J)-K(5,J)*E*C(7,IA,J)+K(2,J)
     .  +2.667*(K(27,J)-C28*K(19,J)+V1)         +4.571*19.0*K(35,J)
      IF(INDEX(1).EQ.2) DCDT(7,L,J)=DCDX(7,J)-K(5,J)*E*C(7,IA,J)+K(2,J)
     .  +2.667*(K(27,J)-C28*K(19,J)+V2)         +4.571*19.0*K(35,J)
      GO TO 121
1003  CONTINUE
      DCDT(1, L,J)=DCDX(1,J)+FRA(1)*K(4,J)-K11*C(1,IA,J)+K(15,J)+F1*K(6,
```

```
     .            J)
      DCDT(2, L,J)=DCDX(2,J)+FRA(2)*K(4,J)+K11*C(1,IA,J)+K(29,J)*E*(CSAT
     .           -C(2,IA,J))+K(16,J)-(C28*K(19,J)+C29*K(20,J)+C210*K(21,
     .            J) +C211*K(22,J)+C212*K(35,J))
      DCDT(3, L,J)=DCDX(3,J)+FRA(3)*K(4,J)-K12*C(3,IA,J)+K(17,J)+F3*K(6,
     .            J)
      DCDT(4, L,J)=DCDX(4,J)+FRA(4)*K(4,J)+K12*C(3,IA,J)-20.0*K(35,J)
      DCDT(5, L,J)=DCDX(5,J)+FRA(5)*K(4,J)+19.0*K(35,J) -(C58*K(19,J)+
     . C59*K(20,J)+C510*K(21,J)+C511*K(22,J)+C512*K(35,J))+F5*K(6,J)
      DCDT(6, L,J)=DCDX(6,J)+FRA(6)*K(4,J)-K36*C(6,IA,J)*C(6,IA,J)+
     .            K(18,J)-(C68*K(19,J)+C69*K(20,J)+C610*K(21,J)+C611*
     .            K(22,J)+C612*K(35,J))+F6*K(6,J)
      DCDT(7, L,J)=DCDX(7,J)-K(5,J)*E*C(7,IA,J)  +   K(2,J)+2.667*(K(27,
     .            J)-C28*K(19,J)+K11*C(1,IA,J))+4.571*19.0*K(35,J)
      DCDT(8, L,J)=DCDX(8,J)+K(19,J)-K(23,J)-K(7,J)-K(11,J)
      DCDT(9, L,J)=DCDX(9,J)+K(20,J)-K(8,J)-K(12,J)
      DCDT(10 ,L,J)=DCDX(10,J)+K(21,J)-K(9,J)-K(13,J)
      DCDT(11, L,J)=DCDX(11,J)+K(22,J)-K(10,J)-K(14,J)
       DCDT(12,L,J)=DCDX(12,J)+K(35,J)-K(33,J)-K(32,J)
   94 CONTINUE
      IA=3-IA
    2 RETURN
      END
```

```
      SUBROUTINE FIRST(Y,DY,DX,NM)
C     SUBROUTINE TO NUMERICALLY EVALUATE FIRST DERIVATIVES.            *
C         Y IS A VECTOR OF VALUES FOR THE INDEPENDENT VARIABLE CORRES- *
C              PONDING TO EQUALLY SPACED VALUES OF THE INDEPENDENT     *
C              VARIABLE(X).                                            *
C         DY IS THE VECTOR OF FIRST DERIVATIVES OF THE DEPENDENT VARI- *
C              ABLE WITH RESPECT TO THE INDEPENDENT VARIABLE(DYDX).    *
C         DX IS THE SPACING BETWEEN SUCCESSIVE VALUES OF THE INDEP.VAR.*
C         NM IS THE LENGTH OF THE VECTORS Y AND DY.                    *
      DIMENSION Y(NM),DY(NM)
      DX60=DX*60.0
      DX12=DX*12.0
      DY(NM)=(Y(NM)-Y(NM-1))/DX
      DY( 2)=(Y( 3)-Y( 1))/(2.0*DX)
      DY(NM-1)=(Y(NM)-Y(NM-2))/(2.0*DX)
      DY( 3)=(-Y( 5)+8.*Y( 4)-8.*Y( 2)+Y( 1))/DX12
      DY(NM-2)=(-Y(NM)+8.0*Y(NM-1)-8.0*Y(NM-3)+Y(NM-4))/DX12
      MM=NM-3
      DO 1 I=4,MM
    1 DY(I)=(Y(I+3)-Y(I-3)-9.*(Y(I+2)-Y(I-2))+45.*(Y(I+1)-Y(I-1)))/DX60
      RETURN
      END
```

```
      SUBROUTINE AREA(IDAY,IZDAY,T,TIDE,FAZ,NM,XAREA,R,WT,A)
C     SUBROUTINE TO CALCULATE ESTUARY CROSS SECTIONAL AREA AS A FUNCT- *
C         ION OF TIDAL PHASE AND POSITION                              *
      DIMENSION XAREA(12,NM),R(38,NM),A(NM)
      DO 10 J=1,NM
      PHASE=(T+(IDAY-IZDAY)*24.0-TIDE-R(38,J))/12.425
      R(31,J)=PHASE-FLOAT(IFIX(PHASE))
      IF(R(31,J).LT.0.0) R(31,J)=1.0+R(31,J)
   10 A(J)=R(3,J)*(1.0+R(36,J)*SIN(WT*(R(31,J)+0.126)))
      RETURN
      END
```

```
      SUBROUTINE VELOC(XPOS,Q,A,WT,NM,R,UA)
C     SUBROUTINE TO CALCULATE FLUID VELOCITY AS A FUNCTION OF FRESH    *
C         WATER FLOW RATE,POSITION,CROSS SECTIONAL AREA,AND TIDAL PHASE*
      COMMON /B12/XQCOEF(10),Z(10),NCOEF,IWK2
      DIMENSION XPOS(NM),A(NM),R(38,NM),UA(NM)
      DO 10 J=1,NM
      XQ=Q*2.445E-08
      DO 20 L=1,NCOEF
      IF(XPOS(J) .GT. Z(L)) XQ=XQ*XQCOEF(L)
 20   CONTINUE
   10 UA(J)=(XQ/A(J)+R(37,J)*SIN(WT*R(31,J)))
      RETURN
      END
```

```
      SUBROUTINE VOLUM(A,NM,UA,Q,WORK,XPOS)
C     SUBROUTINE TO CALCULATE VOLUMETRIC FLOW RATE AS A FUNCTION OF    *
C         FRESH WATER FLOW RATE,POSITION,AND TIDAL PHASE.              *
      COMMON /B12/XQCOEF(10),Z(10),NCOEF,IWK2
      DIMENSION A(NM),UA(NM),WORK(NM),XPOS(NM)
      DO 10 J=1,NM
      XQ=Q*2.445E-08
      DO 20 L=1,NCOEF
      IF(XPOS(J) .GT. Z(L)) XQ=XQ*XQCOEF(L)
 20   CONTINUE
   10 WORK(J)=A(J)*UA(J)-XQ+2.445E-08*Q
      RETURN
      END
```

```
      SUBROUTINE INTP(LN,NP,X,FX,XX,FXX,ACC,NM)
      DIMENSION FY(7), Y(7), X( NM), FX( NM)
      L=0
      K=-1
      MM=1
      DO 90 I=1,NP
      IF(X(I)-XX) 90,91,91
   91 IF(ABS(X(I)-XX)-(ACC+ABS(XX))*ACC)92,92,93
   92 FY(1)=FX(I)
      GO TO 94
   93 J=I
      GO TO 95
   90 CONTINUE
      J=NP
      L=-1
   95 Y(1)=X(J)-XX
      FY(1)=FX(J)
      FYY=FY(1)
   96 IF(L)  97,98,97
   97 J=J+L
      GO TO 99
   98 J=J+K
      IF(K) 1,2,3
    1 XK=-1.
      GO TO 4
    2 XK=0.0
      GO TO 4
    3 XK=1.
    4 K=K-(2*K+XK)
      IF(J-NP)101,101,100
  100 J=J+K
      L=-1
```

```
      GO TO 99
  101 IF(J-1) 102,99,99
  102 J=J+K
      L=1
   99 MM=MM+1
      Y(MM)=X(J)-XX
      FY(MM)=FX(J)
      LL=MM-1
      DO 104 NN=1,LL
  104 FY(MM)=(FY(NN)*Y(MM)-FY(MM)*Y(NN))/(Y(MM)-Y(NN))
      IF(LN-LL) 105,94,105
  105 IF(ABS(FYY-FY(MM))-ACC*(ACC+ABS(FY(MM)))) 94 ,106,106
  106 IF(MM-6) 107,107,94
  107 FYY=FY(MM)
      GO TO 96
   94 FXX=FY(MM)
      RETURN
      END
```

```
C  SUBROUTINE BILDAX  ***** DRAW VERTICAL AND HORIZONTAL AXIS  ***
C $ $ $ $ $ $  $ $ $ $ $ $ $ $ $ $ $ $  $ $ $ $ $ $ $ $$ $ $ $ $ $ $ $ $PLD
C     THIS ROUTINE PLOTS AN AXIS(VERTICAL OR HORIZONAL) WITH TIC MARKS
C     AT EACH GIVEN INTERVAL OR AT 1 INCH INTERVALS, PUTS NUMBERS AT
C     TIC MARK, AND LABELS THE AXIS.
C     A SINGLE CALL DRAWS ONE (X OR Y) AXIS DEPENDING ON 'NT' BELOW.
C
C     CONTROL PARAMETERS
C     X,Y = LOCATION OF GRAPH ORGIN(0.0,0.0) IN INCHES  (REAL)
C     HT = HEIGHT IN INCHES OF THE NUMBERS TO BE PLOTTED ON AXIS (REAL)
C          IF HEIGHT =0.0 PROGRAM WILL SET HEIGHT =0.10
C     FP = BEGINNING NUMBER TO BE PLOTTED AT ORGIN OF AXIS (REAL)
C     ND = NUMBER OF DECIMAL DIGITS TO THE RIGHT OF THE DECIMAL POINT
C          OF NUMBERS TO BE PLOTTED, IF ND=-1 NUMBERS WILL BE PLOTTED
C          WITHOUT A DECIMAL POINT(ONLY DIGITS TO LEFT OF DECIMAL)
C          IF ND=0 PROGRAM WILL SET ND=2 FOR 2 DECIMAL POINTS(INTEGER)
C     VI = VALUE TO INCREMENT NUMBERS BY FOR EACH TIC MARK (REAL)
C     AL = LENGTH OF AXIS IN INCHES    (REAL)
C     ALAB = LABEL TO BE WRITTEN ON AXIS(MAX OF 80 CHARACTERS)
C           THE LABEL FOR AXIS MAY BE PASSED TO THE ROUTINE
C              BY ONE OF THE FOLLOWING METHODS IN THE CALL STATEMENT;;;
C           1. CALL BILDAX(X,Y,HT,FP,ND,VI,AL,ALAB,N4,TC,LD,NT) WHERE
C             ALAB IS A VARIABLE READ INTO STORAGE AS A4 WORDS
C           2. CALL BILDAX(X,Y,HT,FP,ND,VI,AL,'Y-AXIS--',2,TC,LD,NT)
C                 WHERE LABEL IS A LITERAL STRING
C           3. CALL BILDAX(X,Y,HT,FP,ND,VI,AL,8HY-AXIS--,2,TC,LD,NT)
C                 WHERE LABEL IS HOLLERITH STRING
C           THE NINTH ARGUMENT IN THE CALL IS THE TOTAL NUMBER OF
C                LETTERS IN STRING DIVIDE BY 4 PADDED BLANKS MAKE THE
C              LABEL EVENLY DIVISIABLE BY 4). SEE N4 BELOW::
C     N4 = NUMBER OF A4'S ALAB WILL DIVIDE INTO(THE ALAB IS READ
C          IN A4 FORMAT N4 WILL BE THE NUMBER OF A4'S READ MAX 20A4'S
```

```
C     TC = LENGTH IN INCHES BETWEEN TIC MARKS: TC IS SET TO 1.0 BY
C          ROUTINE IF TC WAS CODED =0.0             (REAL)
C     LD = SIDE OF AXIS TIC MARKS WILL APPEAR ON:
C          IF Y-AXIS 0=RIGHT SIDE, 1= LEFT SIDE OF AXIS
C          IF X-AXIS 0=UP, 1= DOWN FROM AXIS
C     NT = TYPE OF AXIS TO BE DRAWN ON THIS CALL::
C          0= HORIZONAL(X), 1=VERTICAL(Y) AXIS TYPE
C
C
      SUBROUTINE BILDAX(X,Y,HT,FP,ND,VI,AL,ALAB,N4,TC,LD,NT)
      DIMENSION ALAB(N4)
      NSTART=0
      A=X
      B=Y
      AT=HT
      IF(AT.LE.0.0)AT=0.10
      BT=AT*1.5
      AC=TC
      IF(AC.LE.0.0)AC=1.0
      AP=FP
      KD=ND
      IF(KD.EQ.0)KD=2
      CN=6./7.
      CL=7.5/7.
      NX=N4*4
      C=AT
      D=-1.0
      WN=AT*CN
      WL=BT*CL
      G=AT/2.
      W=BT/2.
      NMAX=0
```

```
      R=0.0
C
C     DRAW AXIS
C
      IF(NT.EQ.0) GO TO 4
      AY=AL+B
      CALL PLOT(A,AY,2)
      GO TO 6
 4    AX=AL+A
C     AX=AL+A
C     COUNT DIGITS AND DRAWN NUMBERS AND TIC MARKS
C
      CALL PLOT(AX,B,2)
 6    IMAX=IFIX(AP)
      N=1
      L=9
      DO 1 J=1,7
      IF(IMAX.LE.L) GO TO 2
      N=N+1
      L=((L+1)*10)-1
 1    CONTINUE
 2    IF(KD.GT.0) N=N+KD+1
      IF(N.GT.NMAX) NMAX=N
      TN=FLOAT(N)*WN
      AD=0.0
      IF(LD.EQ.1)AD=C
      IF(NT.EQ.0)GO TO 7
      E=D*(TN+AD+AT+X)
      F=B-G
      GO TO 9
    7 E=A-(TN/2.0)
      F=Y+(D*(AD+(2.0*AT)))
```

```
  9   IF(NSTART.EQ.1) GO TO 3
      CALL NUMBER(E,F,AT,AP,0.0,KD)
      NSTART=1
      GO TO 5
  3   CALL PLOT(A,B,3)
      S=C
      IF(LD.EQ.1)S=C*D
      IF(NT.EQ.0) S=S+Y
      IF(NT.EQ.1) S=S+X
      IF(NT.EQ.0) CALL PLOT(A,S,2)
      IF(NT.EQ.1) CALL PLOT(S,B,2)
      CALL NUMBER(E,F,AT,AP,0.0,KD)
  5   IF(NT.EQ.0) A=A+AC
      IF(NT.EQ.1) B=B+AC
      R=R+AC
      AP=AP+VI
      IF(R.LE.AL) GO TO 6
C
C     PUT ON LABELS
C
      AD=0.0
      IF(LD.EQ.1)AD=C
      IF(NT.EQ.0) GO TO 10
      TN=FLOAT(NMAX)*WN
      E=D*(X+TN+AD+(2.0*AT))
      F=Y+((AL-(NX*WL))/2.0)
      CALL SYMBOL(E,F,BT,ALAB,90.0,NX)
      GO TO 11
   10 E=X+((AL-(NX*WL))/2.0)
      F=Y+(D*(AD+(3.0*AT)+BT))
      CALL SYMBOL(E,F,BT,ALAB,0.0,NX)
   11 CALL PLOT(X,Y,3)
```

```
RETURN
END
```

```
      SUBROUTINE CONF(ALPHA,BETA,CCOEF,UCL,LCL,MU,GAMAIN,IFLAG)
      COMMON UL,LL
      REAL LOGAMA,LCL,MU,LL,LCOEF
      I=0
      IFLAG=0
      IF(MU.LE.0)DELTA=.01
      IF(MU.GT.0) DELTA=.1
      X=MU
      UCOEF=(1-CCOEF)/2+CCOEF
      LCOEF=(1-CCOEF)/2
      IF(ALPHA.GT.50) GO TO 31
      GAMA=GAMMA(ALPHA)
      LOGAMA=ALOG(GAMA)
90    CALL  IGAMMA(X,ALPHA,LOGAMA,GAMAIN,IFAULT)
      CHECK=UCOEF-GAMAIN
      IF(ABS(CHECK).LT..0001) GO TO 100
      IF (CHECK.LT.0) GO TO 110
      X=X+DELTA
      GO TO 90
110   DELTA=DELTA/10
      I=1
80    X=X-DELTA
      CALL  IGAMMA(X,ALPHA,LOGAMA,GAMAIN,IFAULT)
      CHECK=UCOEF-GAMAIN
      IF(ABS(CHECK).LT..0001) GO TO 100
      IF(CHECK.GT.0) GO TO 70
      GO TO 80
70    IFLAG=1
100   UCL=X *BETA
      UL=GAMAIN
      X=0.0
      IF(I.EQ.1)DELTA=DELTA*10
```

```
60    CALL IGAMMA(X,ALPHA,LOGAMA,GAMAIN,IFAULT)
      CHECK=LCOEF-GAMAIN
      IF(ABS(CHECK).LT..0001) GO TO 200
      IF (CHECK.LT.0) GO TO 210
      X=X+DELTA
      GO TO 60
210   DELTA=DELTA/10
40    X=X-DELTA
      CALL IGAMMA(X,ALPHA,LOGAMA,GAMAIN,IFAULT)
      CHECK=LCOEF-GAMAIN
      IF(ABS(CHECK).LT..0001) GO TO 200
      IF(CHECK.GT.0)GO TO 50
      GO TO 40
50    IFLAG=1
200   LCL=X*BETA
      LL=GAMAIN
      GAMAIN=UL-LL
      GO TO 33
31    CONTINUE
      C0=2.515517
      C1=0.802853
      C2=.010328
      D1=1.432788
      D2=.189269
      D3=.001308
      SS=ALPHA*BETA*BETA
      SIGMA=SQRT(SS)
      LCOEF=LCOEF*LCOEF
      SLQS=1/LCOEF
      A=ALOG(SLQS)
      T=SQRT(A)
      X=T-(C0+C1*T+C2*T*T)/(1+D1*T+D2*T*T+D3*T*T*T)
```

```
      UCL=MU+SIGMA*X
      LCL=MU-SIGMA*X
      GAMAIN=CCOEF
33    RETURN
      END
```

```
C                                                                     *
C*******************************************************************************
C                                                                     *
C         THE REST OF THE USER INPUT INFORMATION IS TO FOLLOW         *
C              IMMEDIATELY AFTER THIS SUBROUTINE                      *
C $ $ $ $ $ $  $ $ $ $ $ $ $ $ $ $ $ $  $ $ $ $ $ $ $ $$ $ $ $ $ $ $ $ $PLD
C*******************************************************************************
C
      SUBROUTINE IGAMMA(X,P,G,GAMAIN,IFAULT)
      DIMENSION PN(6)
      ACU=1.0E-8
      DFLO=1.0E30
      GIN=0.0
      IFAULT=0
      IF(P .LE. 0.0) IFAULT=1
      IF(X .LT. 0.0) IFAULT=2
      IF(IFAULT .GT. 0 .OR. X .EQ. 0.0) GO TO 50
      FACTOR=EXP(P*ALOG(X)-X-G)
      IF(X .GT. 1.0 .AND. X .GE. P) GO TO 30
      GIN=1.0
      TERM=1.0
      RN=P
   20 RN=RN+1.0
      TERM=TERM*X/RN
      GIN=GIN+TERM
      IF(TERM .GT. ACU) GO TO 20
      GIN=GIN*FACTOR/P
      GO TO 50
   30 A=1.0-P
      B=A+X+1.0
      TERM=0.0
      PN(1)=1.0
```

```
      PN(2)=X
      PN(3)=X+1.0
      PN(4)=X*B
      GIN=PN(3)/PN(4)
   32 A=A+1.0
      B=B+2.0
      TERM=TERM+1.0
      AN=A*TERM
      DO 33 I=1,2
   33 PN(I+4)=B*PN(I+2)-AN*PN(I)
      IF(PN(6) .EQ. 0.0) GO TO 35
      RN=PN(5)/PN(6)
      DIF=ABS(GIN-RN)
      IF(DIF .GT. ACU) GO TO 34
      IF(DIF .LE. ACU*RN) GO TO 42
   34 GIN=RN
   35 DO 36 I=1,4
   36 PN(I)=PN(I+2)
      IF(ABS(PN(5)) .LT. DFLO) GO TO 32
      DO 41 I=1,4
   41 PN(I)=PN(I)/DFLO
      GO TO 32
   42 GIN=1.0-FACTOR*GIN
   50 GAMAIN=GIN
      RETURN
      END
```

```
//GO.SYSIN DD *
 79 0.0      58.56     0.0221200.0  1220.5    69.0       0.0       999        01
000000111111   1
.9    .95
.9    .95
.51   .95
1.1   .95
 .1   .95
.01   .95
.158002     .0550658    -.000457297 -.0000073117.0000007512
7.2211      -.0415148   -.000415578
 2
20.0      2.0
45.0      5.0
```

```
   ORGANIC CARBON
   CARBON DIOXIDE
   ORGANIC NITROGEN
   AMMONIA
   NITRITE+NITRATE
   PHOSPHATE
   OD
   ALGAE
   PROTOZOA
   ZOOPLANKTON
```

```
   HIGHER PREDATORS
   BACTERIA
   52     3     1      CROSS-SECTIONAL AREA
0.000232  0.0        0.000232  1.150779  0.000232  2.301558  0.000232  3.452336
0.000270  4.603116   0.000579  5.753895  0.000309  6.904674  0.000425  8.055452
0.000347  9.206232   0.000347  10.357010  0.000270 11.507780 0.000386  12.658550
0.000386  13.809330  0.000347  14.960120 0.000386  16.1109   0.000772  17.261670
0.000463  18.412460  0.000386  19.563230 0.000965  20.714    0.001120  21.86479
0.000888  23.01556   0.001197  24.16635  0.00112   25.31712  0.001583  26.46789
0.002162  27.61868   0.002317  28.76947  0.002664  29.92024  0.001815  31.07101
0.002471  32.2218    0.001313  33.37257  0.001699  34.52336  0.001815  35.67413
0.002317  36.82492   0.001815  37.97569  0.002896  39.12646  0.001815  40.27725
0.002780  41.42803   0.001506  42.57881  0.001853  43.72958  0.003668  44.88037
0.004286  46.03114   0.004286  47.18193  0.003127  48.3327   0.003707  49.48347
0.004015  50.63426   0.00444   51.78504  0.005483  52.93582  0.005367  54.0866
0.004556  55.23738   0.003784  56.38815  0.004402  57.53894  0.004826  58.68971
    1    28     1      SEWAGE INPUT
                52000.                   0.1
                 1600.                   0.5
                 4400.                 6.906
                 4000.                 6.906
                  240.                 7.482
                 7000.               21.2935
                10000.                 23.02
                 3000.                23.596
                39840.                 23.88
                39400.                24.171
                 1280.                24.171
                 1200.                 24.71
                 5000.                 24.71
                                         -1.
    2     2     1      BENTHAL DEMAND
```

```
          40000.                58.65

 2    6     1     LAND RUNOFF
         100000.                52.00
        1000000.                54.00
         100000.                58.65

 2   36     1     MAXIMUM DEVIATION OF X-AREA
     0.24881E 00          0.69047E 01
     0.21358E 00          0.12659E 02
     0.16069E 00          0.18412E 02
     0.19727E 00          0.24166E 02
     0.39267E 00          0.29920E 02
     0.22857E 00          0.35674E 02
     0.12906E 00          0.41428E 02
     0.92593E-01          0.47182E 02
     0.13583E 00          0.52936E 02
     0.16416E 00          0.58690E 02

52   30     1     HYDRAULIC RADIUS, ESTUARY DEPTH
 2.8871     0.0       13.2874    1.7262    12.0735    2.8769    14.1404    4.0277
21.5879     5.1785    19.0945    6.3293    18.2415    7.4801    19.5866    8.6308
12.9593     9.7816    12.1719   10.9324    10.0066   12.0832     8.0381   13.2339
13.6811    14.3847    11.3517   15.5355    35.2362   16.6863    19.4882   17.8371
11.2205    18.9878    31.3648   20.1386    10.8596   21.2894     8.1693   22.4402
 5.7415    23.5910     3.7402   24.7417     5.1509   25.8925     6.1680   27.0433
 5.5118    28.1941    10.7612   29.3448    12.0407   30.4956    11.6798   31.6464
 9.2192    32.7972     6.0367   33.9480    12.6969   35.0987    28.5105   36.2495
13.9764    37.4003    13.0906   38.5511    14.6325   39.7019    13.6811   40.8526
27.9199    42.0034    14.5013   43.1542    19.0617   44.3050    23.1955   45.4557
23.3924    46.6065    20.1772   47.7573    19.4554   48.9081    10.6299   50.0589
10.7612    51.2097     9.0223   52.3604    11.7126   53.5112    10.8268   54.6620
```

```
  8.4318    55.8128    14.0748   56.9635    13.9108    58.1143    16.1745    59.2651
 80     1     2        ORGANIC CARBON
  4.5673     0.0       8.1113     0.7424     4.8366     1.4848     7.2304     2.2272
  4.8460     2.9696    6.8224     3.7120     5.1097     4.4544     6.3696     5.1968
  5.1424     5.9392    5.5697     6.6816     5.7401     7.4240     5.1129     8.1665
  5.3345     8.9089    4.7722     9.6513     4.4805    10.3937     4.6910    11.1361
  4.3078    11.8785    4.7165    12.6209     4.6585    13.3633     5.2391    14.1057
  5.0527    14.8481    5.3862    15.5905     5.3640    16.3329     5.7524    17.0753
  5.8620    17.8177    6.1146    18.5601     4.9405    19.3025     2.1862    20.0449
  3.8244    20.7873    3.7597    21.5297     4.5968    22.2721     3.8790    23.0145
  5.0151    23.7570    5.3906    24.4994     5.4485    25.2418     4.3750    25.9842
  3.4299    26.7266    3.7039    27.4690     4.1305    28.2114     3.1436    28.9538
  3.0560    29.6962    3.2766    30.4386     4.1643    31.1810     4.1104    31.9234
  4.4384    32.6658    3.9341    33.4082     4.2454    34.1506     4.1219    34.8930
  4.0307    35.6354    3.9925    36.3778     3.8943    37.1202     4.0216    37.8626
  3.8989    38.6051    4.0201    39.3475     3.8286    40.0899     4.1257    40.8323
  3.7253    41.5747    3.4536    42.3171     1.1412    43.0595     0.8118    43.8019
  0.5894    44.5443    1.5042    45.2867     1.8072    46.0291     1.2488    46.7715
  1.5462    47.5139    1.0091    48.2563     0.9055    48.9987     0.6757    49.7411
  0.6505    50.4835    0.8740    51.2259     1.2913    51.9683     1.5393    52.7107
  1.4809    53.4532    1.3124    54.1956     1.0842    54.9380     1.0770    55.6804
  1.0328    56.4228    1.1925    57.1652     1.5567    57.9076     1.7961    58.6500
 80     2     2        CARBON DIOXIDE
  3.2362     0.0       0.7602     0.7424     3.0742     1.4848     0.4329     2.2272
  3.0475     2.9696    0.9118     3.7120     3.7997     4.4544     1.9489     5.1968
  4.8408     5.9392    1.5701     6.6816     4.7213     7.4240     1.5398     8.1665
  4.5481     8.9089    1.3669     9.6513     4.1426    10.3937     1.4634    11.1361
  4.1075    11.8785    2.0525    12.6209     4.5089    13.3633     1.9955    14.1057
  4.2957    14.8481    2.3950    15.5905     4.0915    16.3329     2.6954    17.0753
  4.9500    17.8177    1.5962    18.5601     3.5798    19.3025     0.9100    20.0449
  2.5301    20.7873    1.7552    21.5297     3.2957    22.2721     2.0919    23.0145
  3.0854    23.7570    2.1072    24.4994     2.3971    25.2418     1.3715    25.9842
```

```
  2.0572    26.7266     3.0256    27.4690      3.6163    28.2114      2.5620    28.9538
  2.9554    29.6962     2.8443    30.4386      3.9990    31.1810      3.5215    31.9234
  4.3122    32.6658     2.7488    33.4082      4.3602    34.1506      3.8836    34.8930
  4.1905    35.6354     3.9718    36.3778      4.1082    37.1202      3.9607    37.8626
  4.1626    38.6051     4.1796    39.3475      4.3564    40.0899      4.4210    40.8323
  4.2317    41.5747     3.8115    42.3171      1.5064    43.0595      0.9245    43.8019
  0.7292    44.5443     1.7958    45.2867      2.2150    46.0291      1.5045    46.7715
  1.9815    47.5139     1.2058    48.2563      1.2142    48.9987      0.9218    49.7411
  0.8960    50.4835     1.0235    51.2259      1.3934    51.9683      1.6164    52.7107
  1.6996    53.4532     1.5832    54.1956      1.4873    54.9380      1.2427    55.6804
  1.3642    56.4228     1.3797    57.1652      1.7381    57.9076      1.9758    58.6500
 80    3     2         ORGANIC NITROGEN
  0.4387     0.0        1.3736     0.7424      0.7809     1.4848      1.3797     2.2272
  0.8105     2.9696     1.2772     3.7120      0.7034     4.4544      1.0494     5.1968
  0.5390     5.9392     1.0036     6.6816      0.6386     7.4240      0.9441     8.1665
  0.5999     8.9089     0.9288     9.6513      0.5456    10.3937      0.8874    11.1361
  0.5198    11.8785     0.7762    12.6209      0.4627    13.3633      0.7700    14.1057
  0.4979    14.8481     0.7178    15.5905      0.5429    16.3329      0.7050    17.0753
  0.4953    17.8177     0.8265    18.5601      0.4332    19.3025      0.2640    20.0449
  0.3568    20.7873     0.4524    21.5297      0.3867    22.2721      0.4653    23.0145
  0.5410    23.7570     0.7929    24.4994      0.6033    25.2418      0.6067    25.9842
  0.4113    26.7266     0.3968    27.4690      0.5463    28.2114      0.6146    28.9538
  0.5073    29.6962     0.4851    30.4386      0.3867    31.1810      0.4168    31.9234
  0.3321    32.6658     0.4873    33.4082      0.3004    34.1506      0.3663    34.8930
  0.3115    35.6354     0.3424    36.3778      0.3046    37.1202      0.3432    37.8626
  0.2791    38.6051     0.2911    39.3475      0.2232    40.0899      0.2835    40.8323
  0.2299    41.5747     0.2412    42.3171      0.0464    43.0595      0.0592    43.8019
  0.0341    44.5443     0.0976    45.2867      0.1093    46.0291      0.0874    46.7715
  0.0927    47.5139     0.0864    48.2563      0.0628    48.9987      0.0556    49.7411
  0.0506    50.4835     0.0895    51.2259      0.1438    51.9683      0.1762    52.7107
  0.1500    53.4532     0.1273    54.1956      0.0850    54.9380      0.1246    55.6804
  0.0923    56.4228     0.1330    57.1652      0.1677    57.9076      0.1918    58.6500
```

```
80      4       2          AMMONIA
 0.1421   0.0        0.1995    0.7424    0.4627    1.4848    0.2976    2.2272
 0.5449   2.9696     0.3592    3.7120    0.5625    4.4544    0.3590    5.1968
 0.5611   5.9392     0.3500    6.6816    0.5671    7.4240    0.3828    8.1665
 0.5522   8.9089     0.3782    9.6513    0.5023   10.3937    0.3197   11.1361
 0.4803  11.8785     0.3221   12.6209    0.5773   13.3633    0.4018   14.1057
 0.6812  14.8481     0.4829   15.5905    0.6907   16.3329    0.5386   17.0753
 0.8822  17.8177     0.4078   18.5601    0.6826   19.3025    0.2090   20.0449
 0.5064  20.7873     0.4031   21.5297    0.6142   22.2721    0.4422   23.0145
 0.6021  23.7570     0.5124   24.4994    0.4573   25.2418    0.3126   25.9842
 0.3922  26.7266     0.5444   27.4690    0.7811   28.2114    0.6580   28.9538
 0.6992  29.6962     0.6622   30.4386    0.8649   31.1810    0.7777   31.9234
 0.9118  32.6658     0.6336   33.4082    0.9154   34.1506    0.8581   34.8930
 0.9295  35.6354     0.9050   36.3778    0.9417   37.1202    0.9462   37.8626
 0.9777  38.6051     0.9988   39.3475    1.0255   40.0899    1.0589   40.8323
 1.0044  41.5747     0.9094   42.3171    0.3475   43.0595    0.2172   43.8019
 0.1679  44.5443     0.4164   45.2867    0.5095   46.0291    0.3446   46.7715
 0.4488  47.5139     0.2672   48.2563    0.2638   48.9987    0.1910   49.7411
 0.1836  50.4835     0.2139   51.2259    0.2955   51.9683    0.3454   52.7107
 0.3604  53.4532     0.3312   54.1956    0.3059   54.9380    0.2528   55.6804
 0.2795  56.4228     0.2841   57.1652    0.3689   57.9076    0.4239   58.6500
80      5       2          NITRITE+NITRATE
 0.4764   0.0        0.3498    0.7424    0.5234    1.4848    0.3060    2.2272
 0.5009   2.9696     0.3130    3.7120    0.5549    4.4544    0.3986    5.1968
 0.6602   5.9392     0.3704    6.6816    0.6600    7.4240    0.3666    8.1665
 0.6376   8.9089     0.3338    9.6513    0.5753   10.3937    0.3142   11.1361
 0.5463  11.8785     0.3518   12.6209    0.6051   13.3633    0.3894   14.1057
 0.6320  14.8481     0.4561   15.5905    0.6196   16.3329    0.4825   17.0753
 0.6962  17.8177     0.3834   18.5601    0.5102   19.3025    0.1665   20.0449
 0.3759  20.7873     0.3118   21.5297    0.4768   22.2721    0.3545   23.0145
 0.4636  23.7570     0.3611   24.4994    0.3316   25.2418    0.2097   25.9842
 0.2562  26.7266     0.3890   27.4690    0.4498   28.2114    0.3093   28.9538
```

```
  0.3466    29.6962     0.3485   30.4386     0.4787   31.1810     0.4344   31.9234
  0.5151    32.6658     0.3998   33.4082     0.5396   34.1506     0.5250   34.8930
  0.5339    35.6354     0.5211   36.3778     0.5170   37.1202     0.5258   37.8626
  0.5247    38.6051     0.5479   39.3475     0.5556   40.0899     0.6023   40.8323
  0.5635    41.5747     0.5347   42.3171     0.1860   43.0595     0.1337   43.8019
  0.1004    44.5443     0.2563   45.2867     0.3093   46.0291     0.2170   46.7715
  0.2673    47.5139     0.1796   48.2563     0.1606   48.9987     0.1244   49.7411
  0.1207    50.4835     0.1604   51.2259     0.2351   51.9683     0.2810   52.7107
  0.2771    53.4532     0.2490   54.1956     0.2127   54.9380     0.2020   55.6804
  0.2070    56.4228     0.2321   57.1652     0.3071   57.9076     0.3558   58.6500
 80     6    2         PHOSPHATE
  0.2408     0.0        0.7261    0.7424     0.8625    1.4848     0.8341    2.2272
  0.9403     2.9696     0.8195    3.7120     0.8562    4.4544     0.6784    5.1968
  0.7287     5.9392     0.6334    6.6816     0.7048    7.4240     0.6134    8.1665
  0.6009     8.9089     0.5524    9.6513     0.4769   10.3937     0.4537   11.1361
  0.4238    11.8785     0.4415   12.6209     0.5468   13.3633     0.5961   14.1057
  0.6829    14.8481     0.6284   15.5905     0.6580   16.3329     0.5949   17.0753
  0.7183    17.8177     0.5415   18.5601     0.5663   19.3025     0.2176   20.0449
  0.4343    20.7873     0.4028   21.5297     0.5202   22.2721     0.4181   23.0145
  0.5701    23.7570     0.5515   24.4994     0.3704   25.2418     0.2798   25.9842
  0.3002    26.7266     0.4339   27.4690     0.5772   28.2114     0.4533   28.9538
  0.4547    29.6962     0.4397   30.4386     0.5452   31.1810     0.4881   31.9234
  0.5418    32.6658     0.3895   33.4082     0.5270   34.1506     0.4846   34.8930
  0.5067    35.6354     0.4814   36.3778     0.4825   37.1202     0.4566   37.8626
  0.4672    38.6051     0.4605   39.3475     0.4630   40.0899     0.4670   40.8323
  0.4388    41.5747     0.3882   42.3171     0.1504   43.0595     0.0898   43.8019
  0.0699    44.5443     0.1715   45.2867     0.2110   46.0291     0.1401   46.7715
  0.1872    47.5139     0.1077   48.2563     0.1098   48.9987     0.0788   49.7411
  0.0759    50.4835     0.0852   51.2259     0.1163   51.9683     0.1346   52.7107
  0.1405    53.4532     0.1286   54.1956     0.1187   54.9380     0.0957   55.6804
  0.1036    56.4228     0.1019   57.1652     0.1285   57.9076     0.1464   58.6500
```

```
   16.    10350.          450.   120                           1
          8000.           600.   122                           1
          8750.           597.   121                           1
//GO.FT08F001 DD *

//GO.FT09F001 DD *
 &LIST1 PRIE=.01,CL=1.0 ,FRA=.44,.10,.164,.0915,.0175,.181,X11=.10,X36=0.10,
  BA=0.16,D1=.20,GRO(1)=.0015,X14=.6   ,KM12=.50E07,PRED(1)=.0010,D2=.40,
 CK3=1.0,BC(1)=0.05,BC(5)=0.10,DEV=11.0,  &END
```

APPENDIX IV

GRAPHS

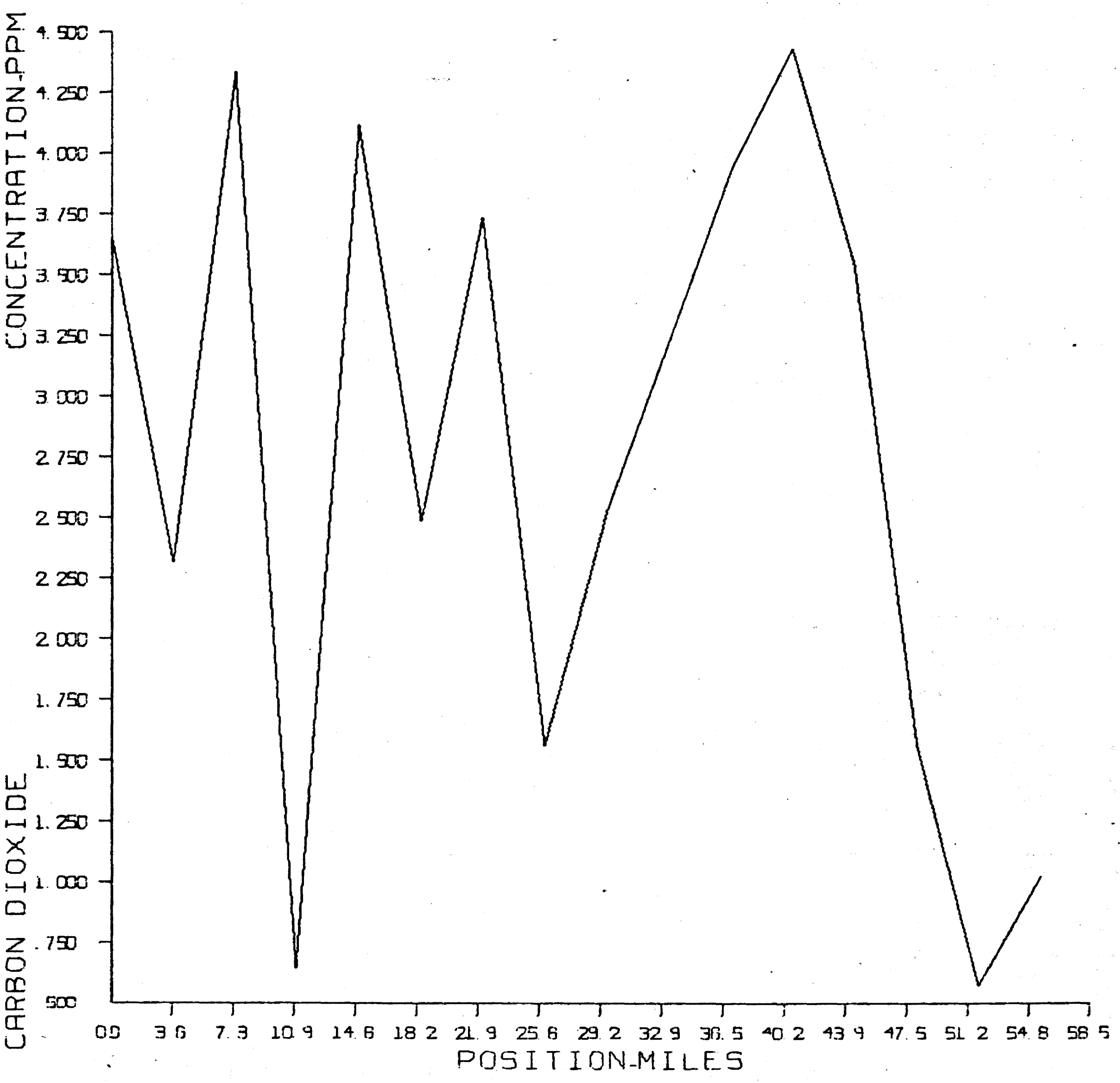


FIGURE 1

Day 1

Day 2

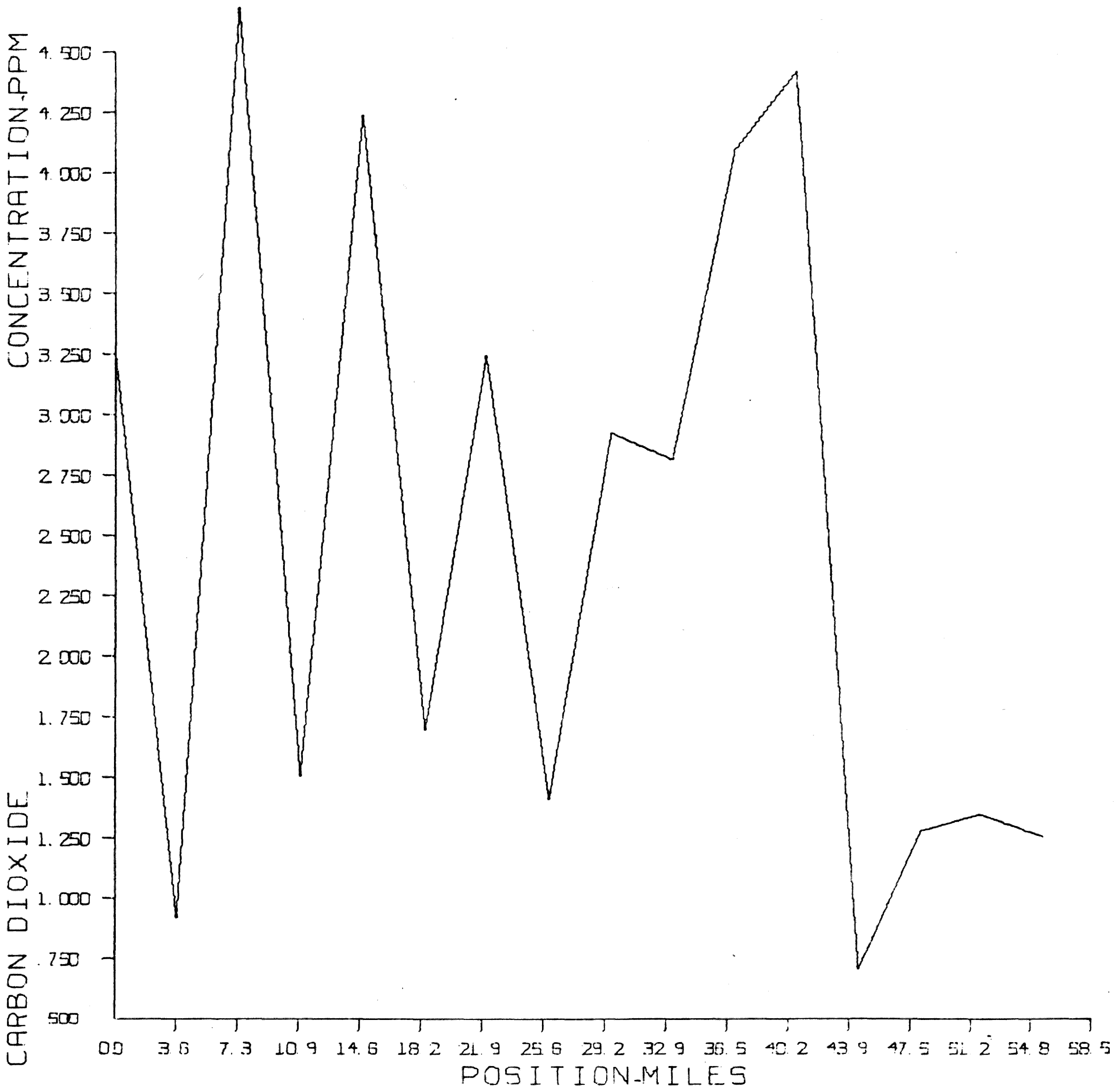
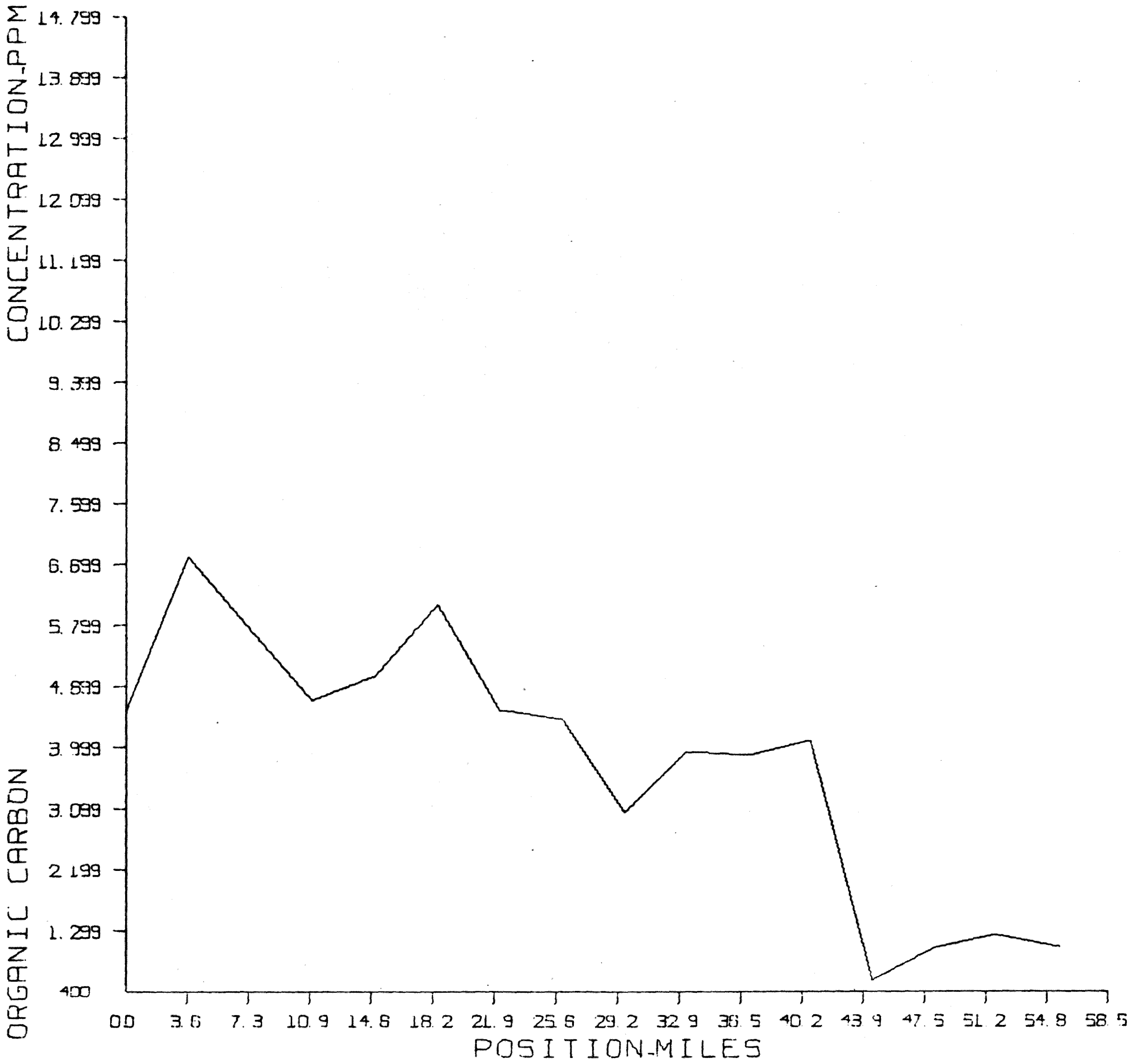
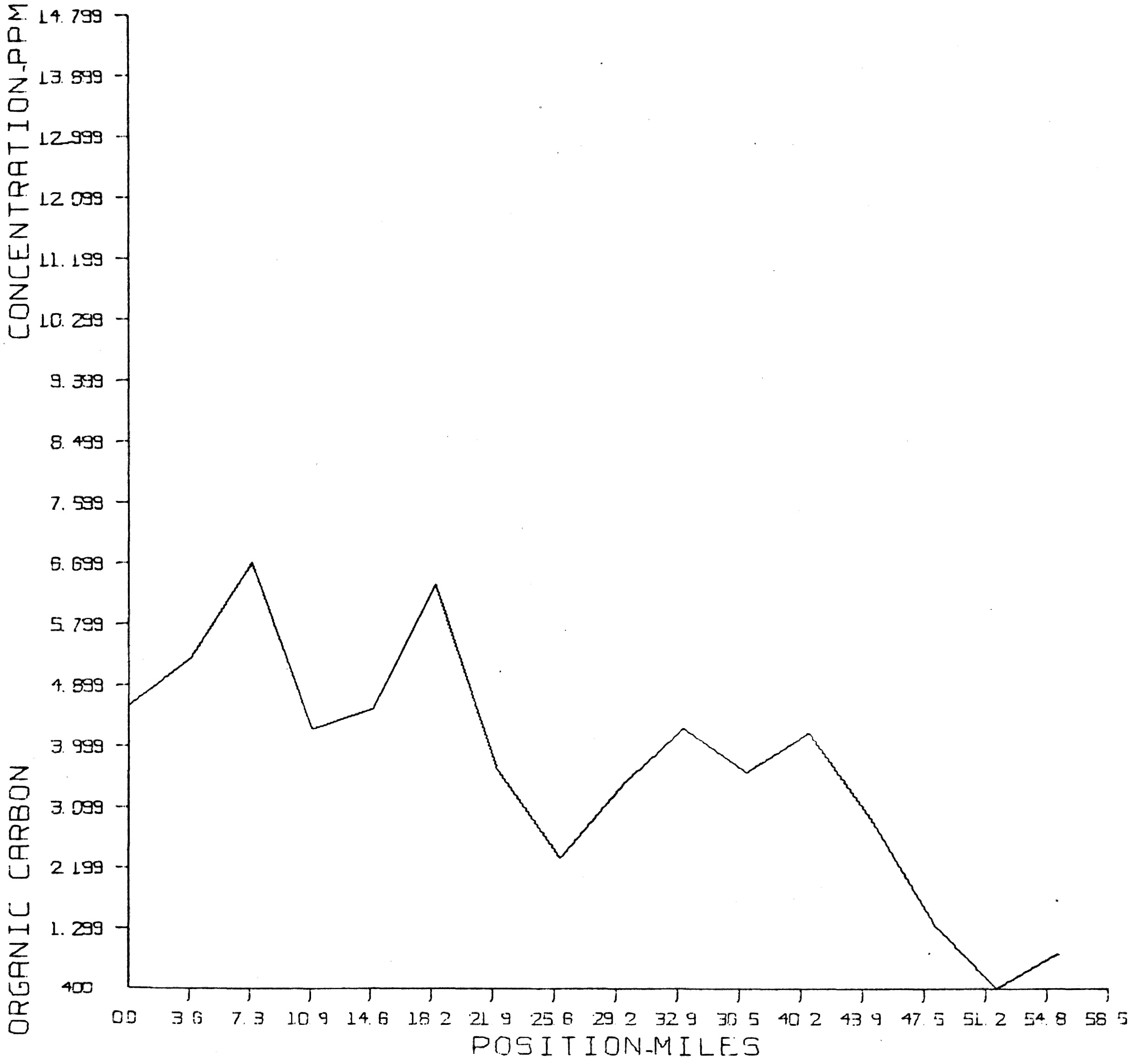


FIGURE 2



Day 1

FIGURE 3



Day 2

FIGURE 4

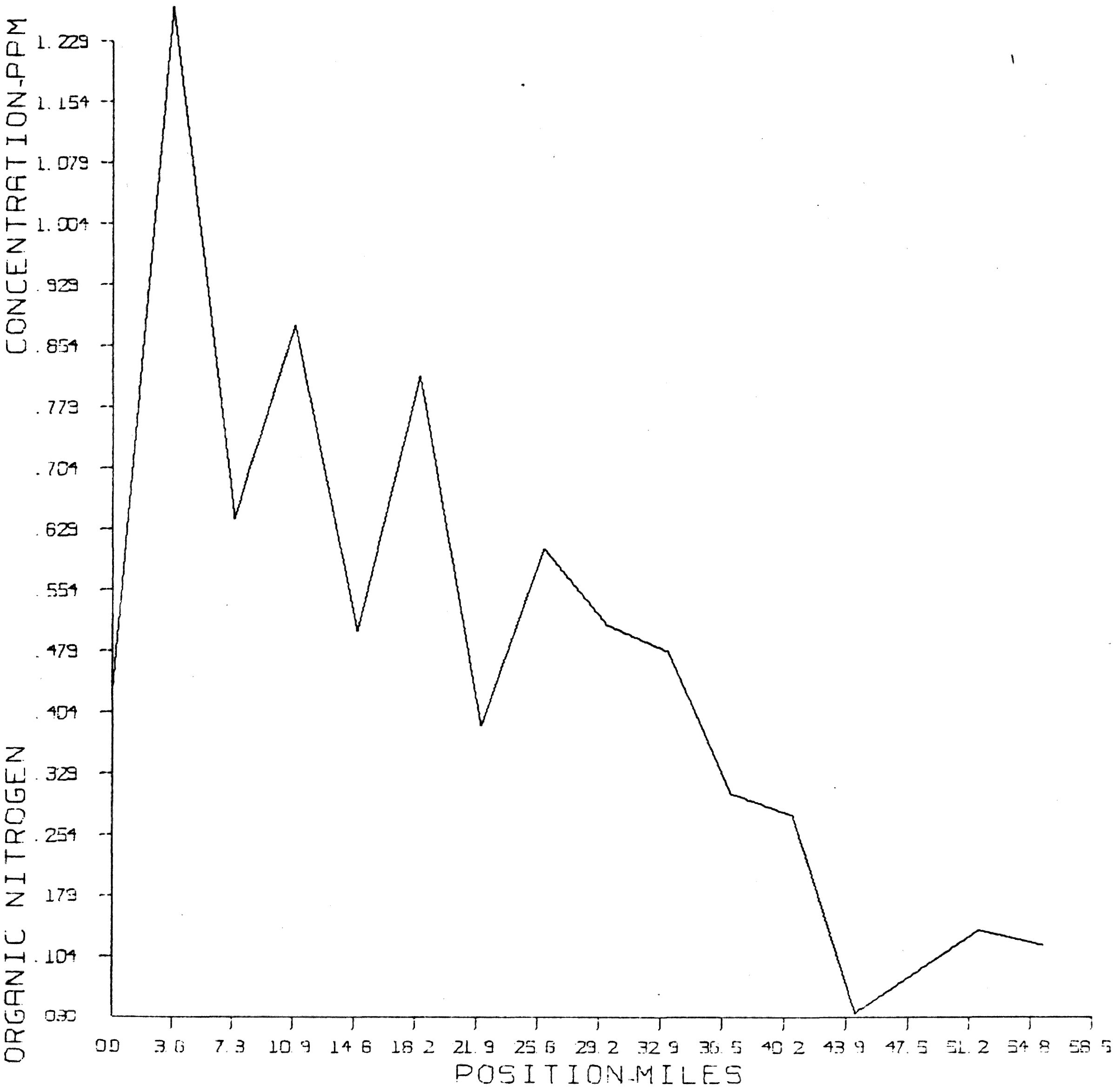

ORGANIC NITROGEN CONCENTRATION-PPM
1.229
1.154
1.079
1.004
.929
.854
.779
.704
.629
.554
.479
.404
.329
.254
.179
.104
0.30
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 5

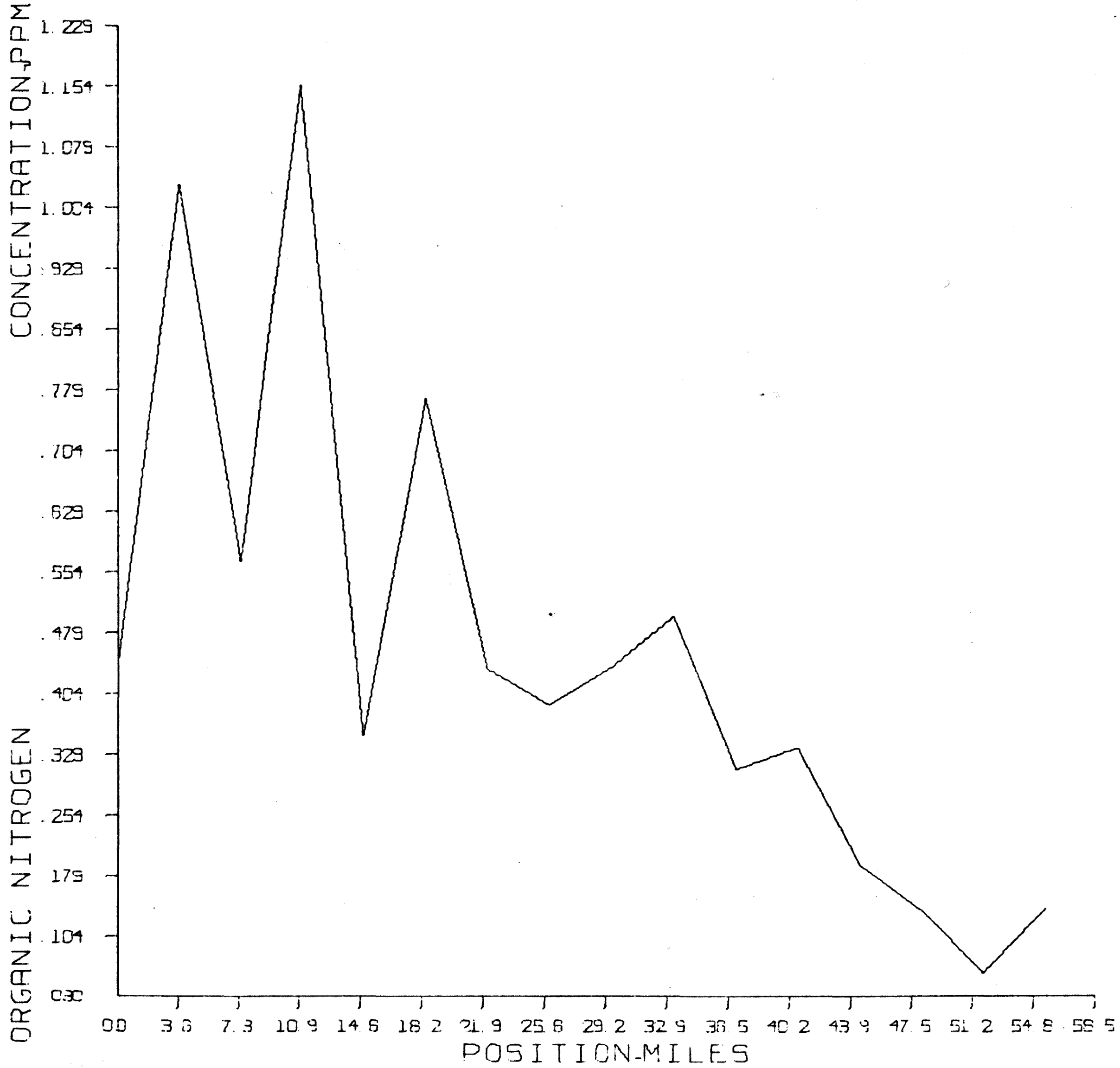

ORGANIC NITROGEN CONCENTRATION,PPM
1.229
1.154
1.079
1.004
.929
.854
.779
.704
.629
.554
.479
.404
.329
.254
.179
.104
0.00
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 6

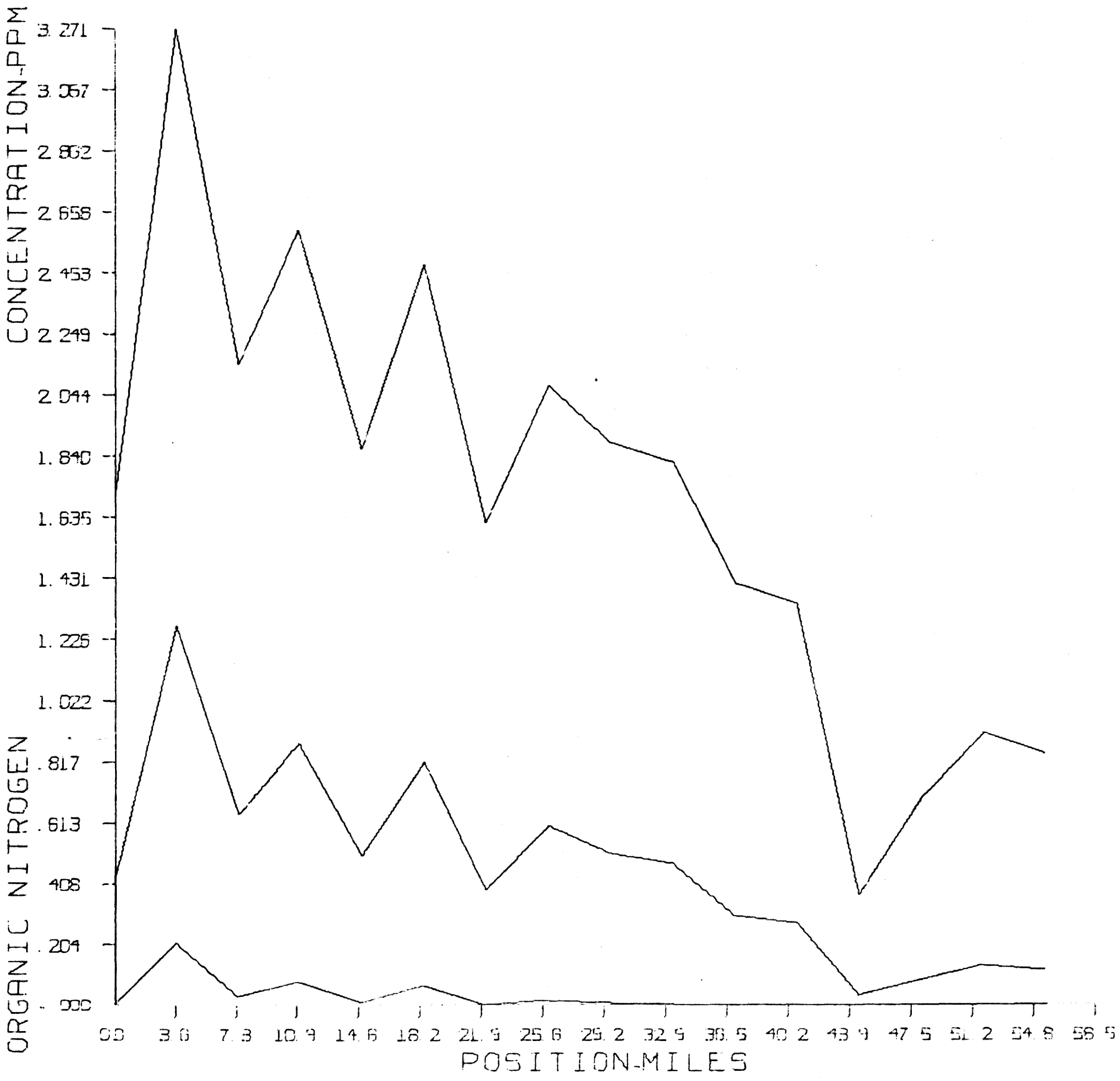

ORGANIC NITROGEN CONCENTRATION-PPM
3.271
3.067
2.862
2.658
2.453
2.249
2.044
1.840
1.635
1.431
1.226
1.022
.817
.613
.408
.204
.000
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 7

FIGURE 8

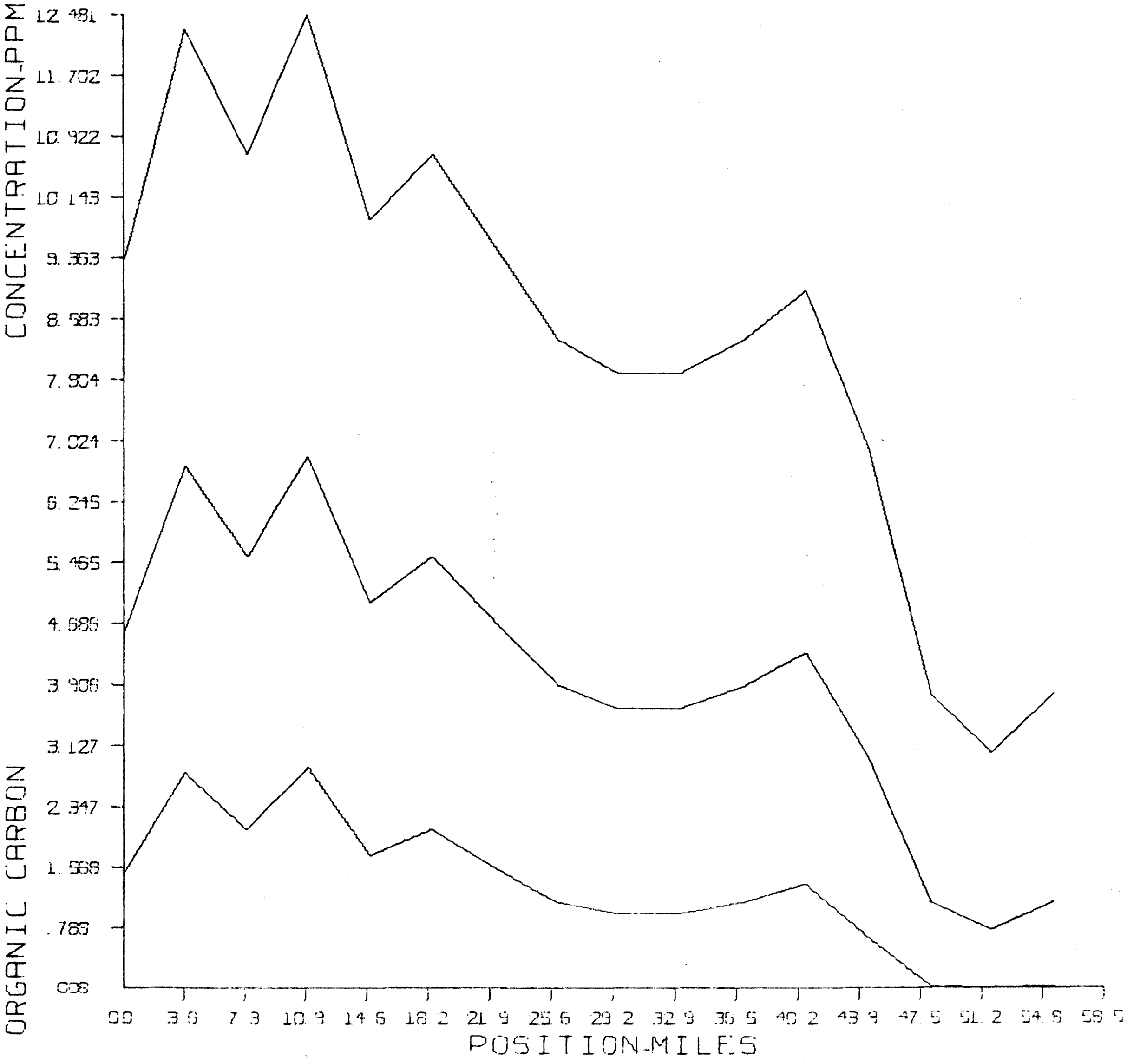
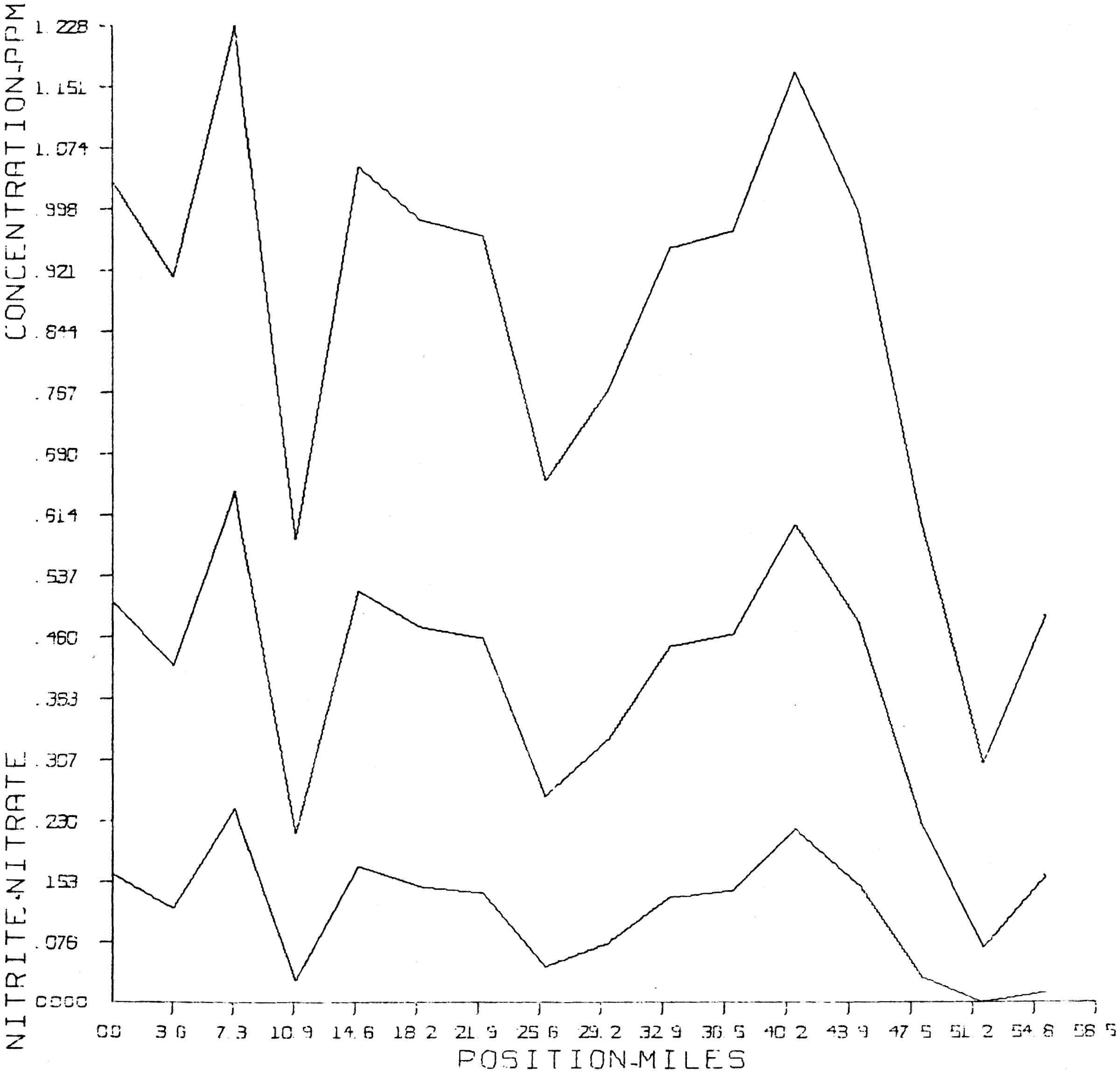

ORGANIC CARBON CONCENTRATION-PPM
12.481
11.702
10.922
10.143
9.363
8.583
7.804
7.024
6.245
5.465
4.686
3.906
3.127
2.347
1.568
.788
.008
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 9



NITRITE-NITRATE CONCENTRATION-PPM
1.228
1.151
1.074
.998
.921
.844
.767
.690
.614
.537
.460
.383
.307
.230
.153
.076
0.000
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

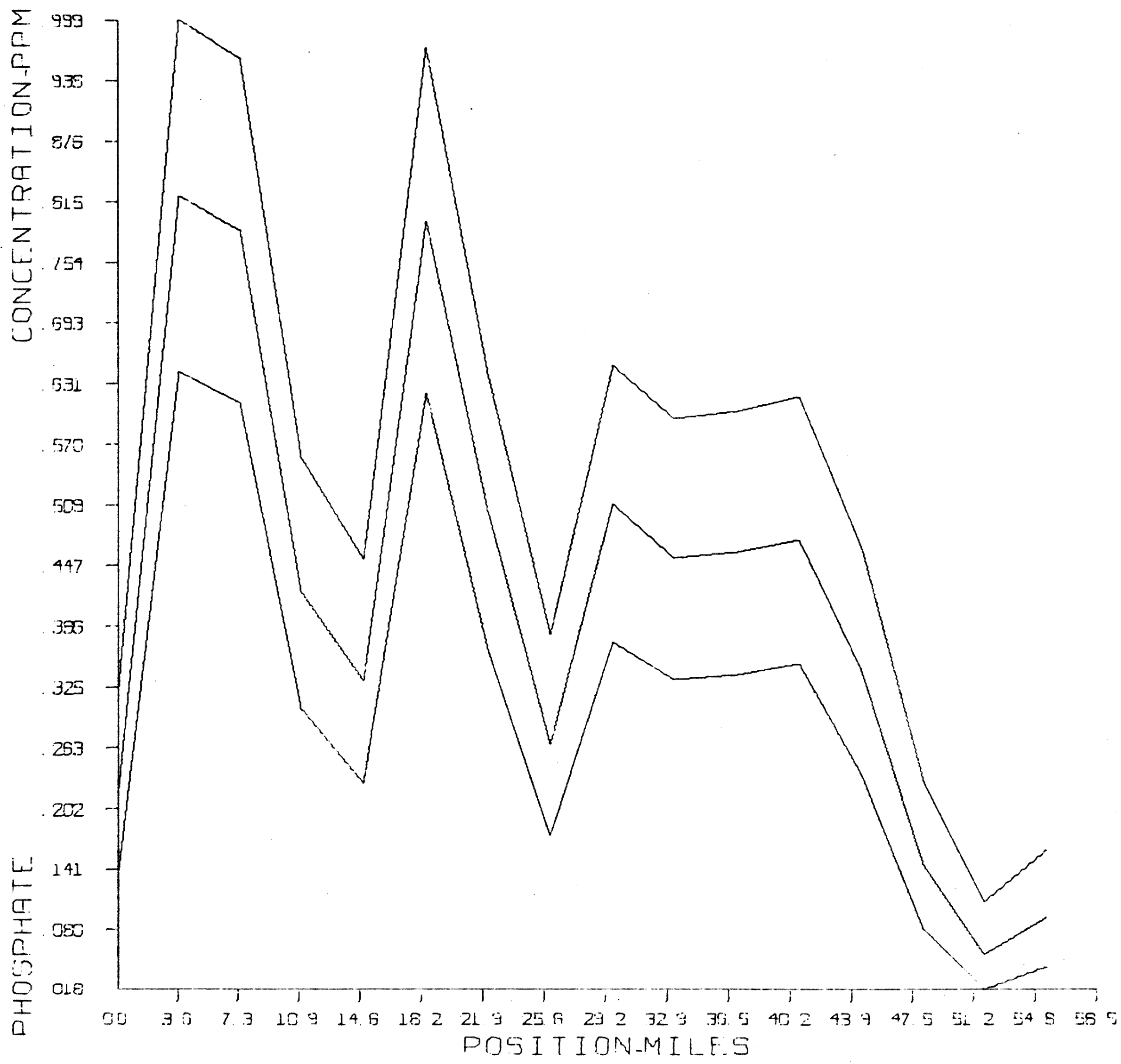

PHOSPHATE CONCENTRATION-PPM
.999
.938
.876
.815
.754
.693
.631
.570
.509
.447
.386
.325
.263
.202
.141
.080
.018
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 10

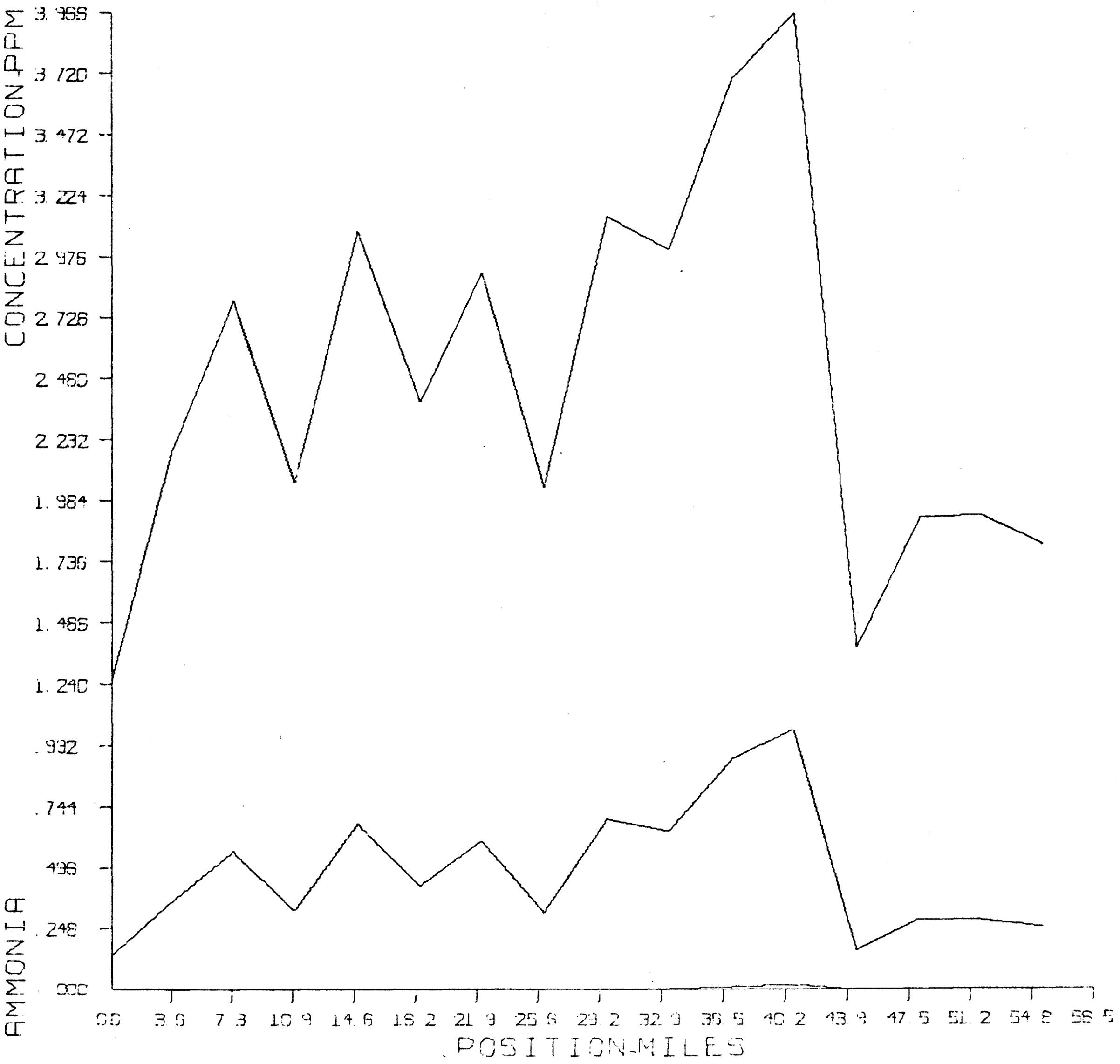

AMMONIA CONCENTRATION,PPM
3.968
3.720
3.472
3.224
2.976
2.728
2.480
2.232
1.984
1.736
1.488
1.240
.992
.744
.496
.248
.000
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.8
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 11

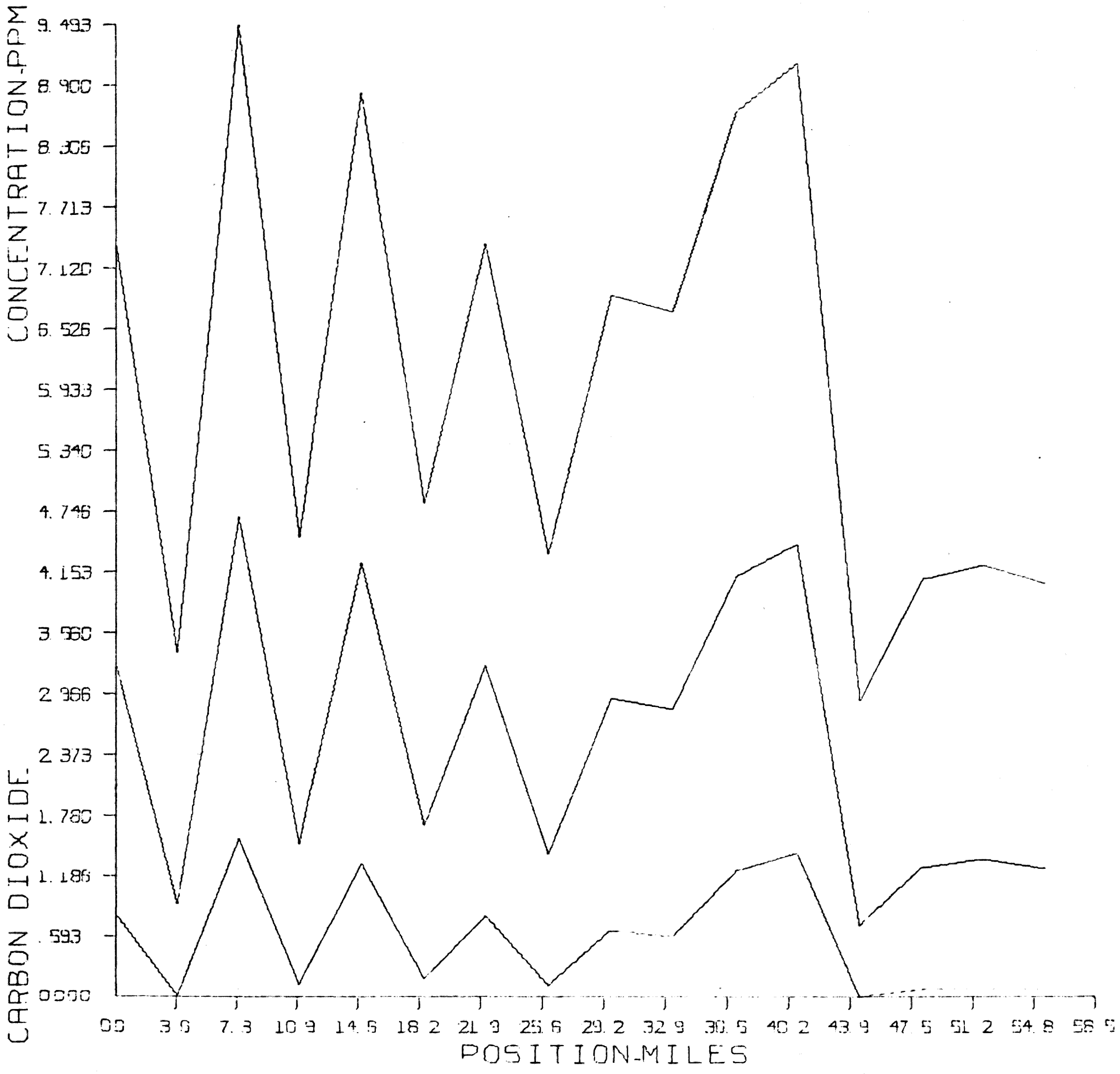

CARBON DIOXIDE CONCENTRATION-PPM
9.493
8.900
8.305
7.713
7.120
6.526
5.933
5.340
4.746
4.153
3.560
2.966
2.373
1.780
1.186
.593
0.000
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

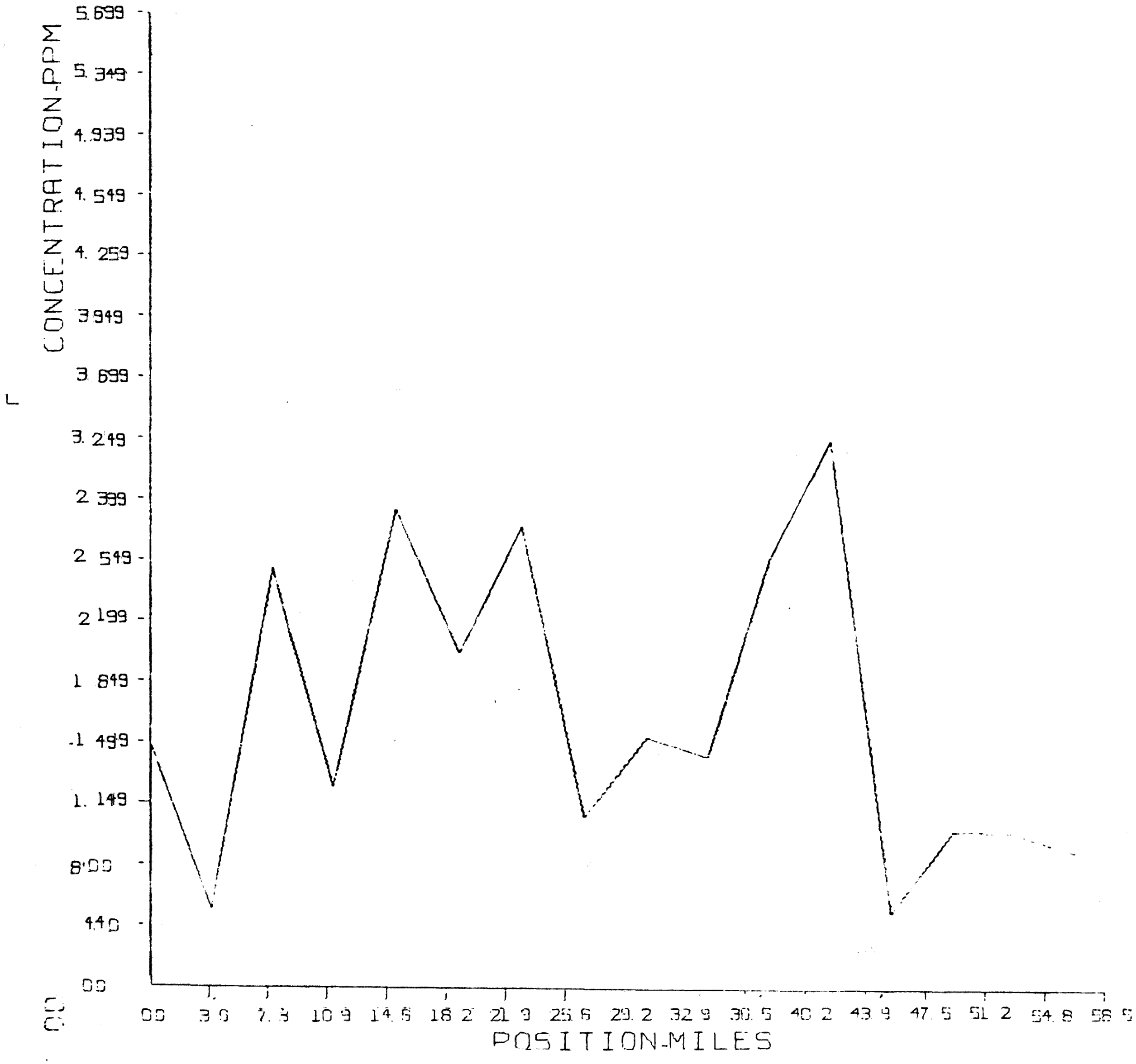
FIGURE 12

The next two graphs illustrate the estimated concentration of oxygen deficit obtained from the full model and from a reduced model. The components in the reduced model in this case are organic carbon, carbon dioxide, algae and oxygen deficit. These were chosen as they are the four most frequently used components and a thorough study of all possible subsets of the twelve original components is not feasible at this time. In order to reduce the model to these four, default values in ppm, that are internal to the program, were assigned to the remaining eight components. Two runs were made under identical estuary conditions: one for the reduced model and one for the full model. The first graph depicts the oxygen deficit concentration as estimated for the full model and the second depicts that for the reduced model.

Initially the two concentrations are similar, but they tend to diverge with increased time and distance. In the full model each concentration is changing and affecting the others, whereas in the reduced model only four are free to interact while the rest are held to constant values.

If actual data were available, a more thorough study of the effects of using a variety of reduced models could be made by comparing the results obtained by each to the true values instead of to those obtained through the use of the full model. Addition information could be obtained by varying the default values used and again comparing the results to the actual values.

FIGURE 13


CONCENTRATION-PPM
5.699
5.349
4.939
4.549
4.259
3.949
3.699
3.249
2.399
2.549
2.199
1.849
1.499
1.149
8.00
4.40
0.0
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
30.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

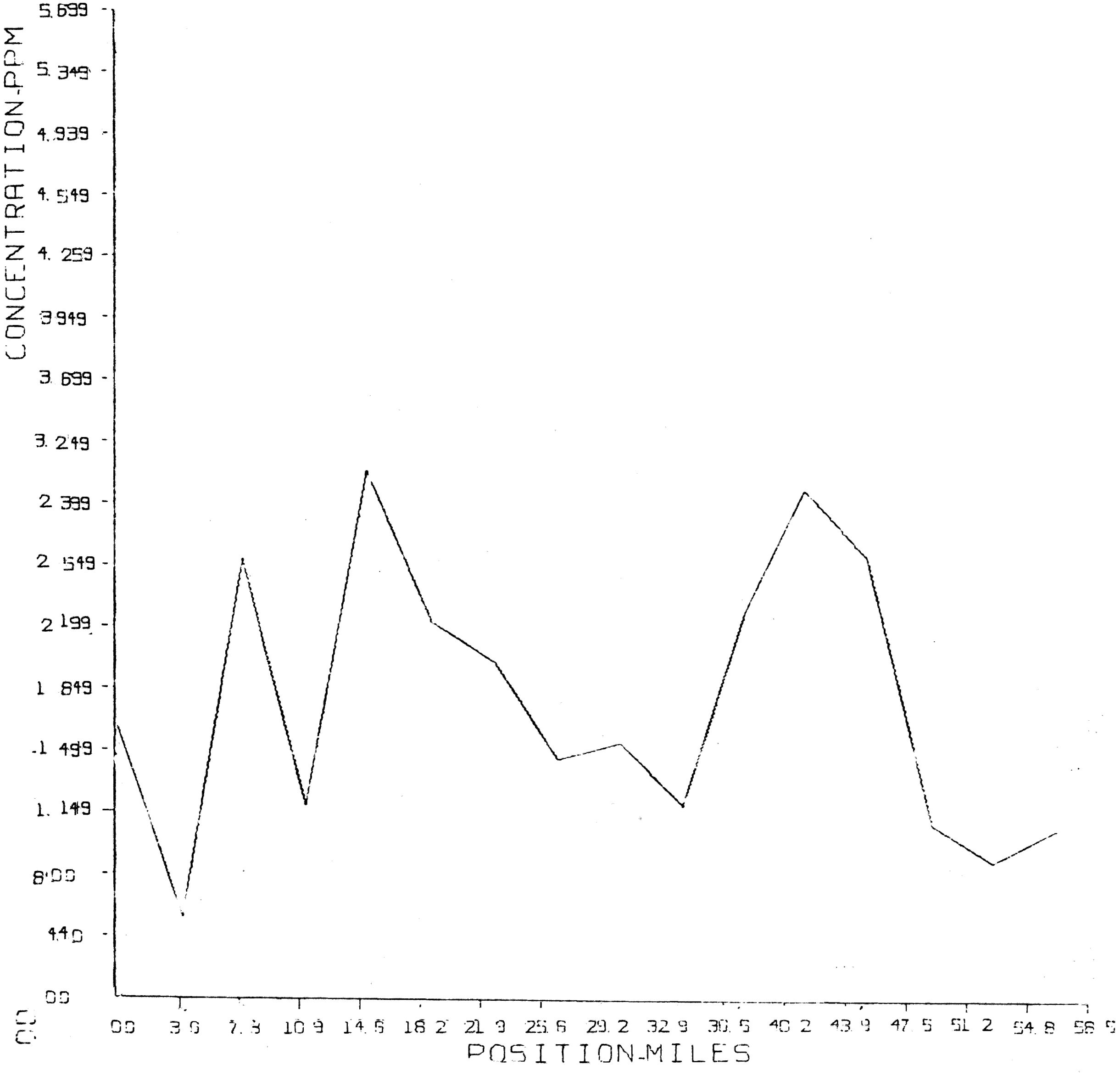

CONCENTRATION-PPM
5.699
5.349
4.939
4.549
4.259
3.949
3.699
3.249
2.399
2.549
2.199
1.849
1.499
1.149
8.00
4.40
0.0
0.0
3.6
7.3
10.9
14.6
18.2
21.9
25.6
29.2
32.9
36.5
40.2
43.9
47.5
51.2
54.8
58.5
POSITION-MILES

FIGURE 14

MODIFICATIONS ON THE SCHOFIELD STOCHASTIC MODELS FOR ESTUARIES

by

Nicholas John Manthos

(ABSTRACT)

Now with the continuing development of computers, more detailed and complex mathematical models are being developed for the use of modeling estuaries. Schofield and Krutchkoff developed a one-dimensional stochastic model to predict the concentration of twelve components, five biological and seven chemical.

It was criticized because all the components had to be included in the model in order for its operation when in fact one only wanted to study a subset of them. It was the object of this work to reduce the restrictions that all twelve had to be included and make it able to use any subset of the twelve that are wished, in such a way that with limited experience in computer programming was sufficient to operate the program.

Also more options as to what type of graphs, and initial run conditions were added to make use easier, as well as the addition of confidence intervals for each component concentration.

It is now hoped that with these simplifications and added options that this model can now become a viable tool in the field of water pollution.